On Modifications to the Traffic-Responsive Urban Control Method

Von der
Fakultät Architektur, Bauingenieurwesen und
Umweltwissenschaften
der Technischen Universität Carolo-Wilhelmina
zu Braunschweig

zur Erlangung des Grades eines
Doktoringenieurs (Dr.-Ing.)
genehmigte

Dissertation

von
Luciano Dionisio Dantas
geboren am 26.06.1978
aus Aracaju – Brasilien

Eingerichtet am 10 Januar 2014
Disputation am 2 September 2014

Berichterstatter Prof. Friedrich
Prof. Schnieder

2014

Acknowlegments

First of all, I would like to thank my partner for all the support that she has given me. Without it, I don't believe that I would have made it. My family's support has also been invaluable, fostering me in the achievement of my goals that they also see as best for me.

Thank you Rodrigo Castelan Carlson for the constructive critiques that were essential in making the present work better.

Thank you Prof. Werner Kraus Junior for all your time and support given throughout these years.

Many thanks for the whole team of the *Institut für Verkehr und Stadtbauwesen* and Prof. Bernhard Friedrich.

Finally, I would also like to thank the CAPES Foundation, Ministry of Education of Brazil, for the scholarship that made the present work possible.

Abstract

The current work has its focus on further improvements envisioned for an existing traffic control system called Traffic-responsive Urban Control (TUC). Originally conceived for corridor networks, TUC only offers the possibility to maintain synchronized traffic lights that give right-of-way for the vehicles traveling through the main routes or, more specifically, the routes that do not intersect. This synchronization is achieved through the adjustment of the Offset parameter, and it is known to avoid unnecessary stops at the successive traffic controlled intersections, reducing traffic delays and increasing the drivers' comfort.

The present investigation proposes an extension to TUC's original formulation, enabling it to handle more complex networks (meshed networks), where the secondary intersecting routes may also profit from traffic lights synchronization. Moreover, TUC's original method, employed during the necessary changes in Offsets, is also improved. The new method takes into consideration the impacts that the change in Offsets may incur to the operation of the network.

TUC's main input information, during its operation, is the description of current traffic queue lengths. Complementing the mentioned modifications to TUC, a new method for the estimation/prediction of traffic queues is presented. The proposed Queue Estimator/Predictor uses a macroscopic traffic model to capture the traffic dynamics of the network, and uses this information for improving the traffic queue estimations calculated in a previous step.

Finally, the evaluation of the current developments is presented. The evaluation is carried out through the simulation of a real network during a whole day operation. The new developments are not only compared to TUC's original formulation, but also against a recently developed Adaptive Traffic Control System (ATCS) prototype. The results show that the developments proposed in the current work were indeed beneficial to TUC's operation, even though the improvements were not quite as high as expected.

Kurzfassung

Das Verkehrs-Steuersystem *Traffic-responsive Urban Control* (TUC) wurde ursprünglich konzipiert für Hauptverkehrsadern, und bietet nur die Möglichkeit, die Schaltzeit der Lichtsignalanlagen zu synchronisieren, die die Durchfahrt der Hauptvekehrsströme vorberechtigen. Nebenströmen , die die Hauptroute überschneiden, können durch TUC in der Regel nicht synchronisiert werden. Diese Synchronisierung wird durch die Einstellung der Versatzzeit erreicht, und sie vermeidet die unnötige Stopps an aufeinanderfolgenden signalgesteuerten Knotenpunkten. Dadurch werden Verzögerungen im Verkehrsablauf reduziert und der Komfort der Fahrer erhöht.

Die vorliegende Arbeit schlägt eine Erweiterung der ursprünglichen Formulierung TUCs vor, wodurch komplexe Netzwerke behandelt werden können, und auch die sekundären Nebenströmen von der Synchronisierung profitieren. Darüber hinaus ist die ursprüngliche Methode TUC, die für die notwendigen Veränderungen in Versatzzeiten verwendet wird, auch verbessert. Die neue Methode berücksichtigt die Auswirkungen auf den Betrieb des Verkehrsnetzes, die die Änderung der Versatzzeiten ergeben.

Eine der wichtigsten Eingangsdaten TUCs ist die Beschreibung der aktuellen Rückstaulängen. Ergänzend zu den oben genannten Änderungen zu TUC, eine neue Methode für die Schätzung/Prognose von Rückstaus wird vorgestellt. Die vorgeschlagene Rückstauschätzer-prädiktor verwendet ein makroskopisches Verkehrsmodell, um die Verkehrsdynamik des Netzes zu erfassen. Diese Dynamik wird dann benutzt, um die Schätzung der Rückstaulängen, die in einem vorherigen Schritt berechnet wurden, zu verbessern.

Schließlich ist die Beurteilung der aktuellen Entwicklung dargestellt. Die Auswertung erfolgt durch die Simulation eines realen Netz während eines ganzen Tages. Die neuen Entwicklungen sind nicht nur mit der ursprünglichen Formulierung TUCs verglichen, sondern auch gegen eine andere kürzlich entwickelte *Adaptive Traffic Control System* (ATCS) Prototyp. Die Ergebnisse zeigen, dass die vorgeschlagene Entwicklungen in Vorteil für TUC waren, obwohl die Verbesserungen nicht ganz so hoch wie erwartet waren.

Contents

1 Introduction 1
- 1.1 Objectives 3
- 1.2 Outline 4
- 1.3 Basic Concepts and Terminology 4

2 Literature Review 7
- 2.1 Adaptive Traffic Control Systems 7
  - 2.1.1 SCOOT 7
  - 2.1.2 SCATS 9
  - 2.1.3 BALANCE 11
  - 2.1.4 OPAC 12
  - 2.1.5 RHODES 13
  - 2.1.6 Pohlmann's 15
  - 2.1.7 TUC 17

3 TUC 19
- 3.1 Control Scheme 19
- 3.2 Cycle Control Module 20
  - 3.2.1 Webster's Cycle Alternative 21
  - 3.2.2 Saturation Based Cycle Alternative 23
- 3.3 Offset Control Module 24
- 3.4 Green Control Module 28
  - 3.4.1 Traffic Flow Model 28
  - 3.4.2 Control Problem 30

4 Proposed Modifications to TUC 39
- 4.1 Proposed Alternative 40
  - 4.1.1 Green Control Module Extension 40
  - 4.1.2 Offset Module 46
  - 4.1.3 Unexpected Setback 52
- 4.2 Legacy Alternative 55

5 Queue Estimation 57
5.1 Local Queue Estimation 59
5.1.1 Mück's Local Queue Estimator 59
5.1.2 Vigos's Local Queue Estimator 60
5.2 Cell Transmission Model 62
5.3 Unscented Kalman Filter 66

6 Evaluation 71
6.1 System Setup 71
6.1.1 Test Network 71
6.1.2 Simulation Environment 72
6.1.3 External Traffic Control Module 74
6.2 Queue Predictor Evaluation 75
6.3 Cycle Alternatives Evaluation 80
6.4 Offset Alternatives Evaluation 83

7 Conclusion 93
7.1 Summary 93
7.2 Discussion 95

A Appendix 97
A.1 LQR 97
A.2 Cell Transmission Model 98
A.2.1 Merges 99
A.2.2 Diverges 101
A.3 Standard Kalman Filter 101
A.4 Unscented Kalman Filter 103
A.5 Performance Comparison of the Statistical Indicators Used 108
A.6 Tests with Pohlmann's ATCS 109

Bibliography 113

List of Symbols 123

List of Figures 127

List of Tables 129

List of Algorithms 129

1 | Introduction

In its inception, urban traffic control had only one goal: to guarantee an ordered and safe operation of traffic by outlining the rules and conventions that dictate how to behave in traffic. Along with its evolution came the standardization of road signs and pavement markings, and the development of automated devices for directing traffic in intersections. The increase of traffic in the cities has forced the urban traffic control to shift its efforts for improving the operation of traffic by trying to optimize the amount of green time given to each traffic stream, therefore deriving a new ramification: traffic signal control. Enabled by the progress in technology, and thereafter the increase in computation power, there has been a continuous improvement and development of traffic control systems that seek a better management of traffic by responding to, and anticipating the oscillations in traffic conditions. In this process, there are three main fronts that can be identified. The first one is the correct estimation of current traffic conditions, including the estimation of queue lengths, vehicle flows and and routes chosen by the drivers traveling through the traffic network. The second front, closely dependent on the first, is the correct assessment of future developments of traffic conditions, which involve the same variables. The third front, given the information gathered/processed on the other two, is responsible for the definition of the actual traffic control actions, i.e. the definition of the green light lengths that will render the best operation of traffic based on pre-determined performance indexes like delays, travel times, number of stops, etc.

Even with the evolution of traffic signal control and the development of systems that are able to react in real time to changes in traffic conditions (in the network as a whole), the previous and much more simple solutions are still employed. Fixed time signal control is the oldest and most employed of them, given its cheaper price and lowest complexity. It consists of the use of fixed green times, previously stipulated, that are followed by the traffic controller device during the course of the day or a given timeframe. The actual improvement in this type of control method is seen in the computerized tools used to aid the traffic planner in defining these fixed green times, like

the renowned TRANSYT[1] software tool.

The other type of traffic control that still widely used is the actuated traffic control, which emerged with the advent of more powerful processing microcontrollers used in the traffic controller devices. This type of control uses the "raw" information collected from vehicle detectors (e.g. vehicle count; occupancy; headway – time interval between detections) to decide whether to increase green times or not for a specific traffic stream in an intersection. The decision is based on simple rules and usually just compare the measured values with pre-specified thresholds. The resultant impact of the taken actions is completely disregarded either in relation to the intersection in question or the neighbouring ones.

The last category of traffic control, the so called Advanced Traffic Control Systems (ATCS), involves the use of traffic flow models to evaluate and optimize the traffic control actions that will be implemented according to current traffic conditions. In this type of system, there is always some sort of centralized processing entity where the whole network is accounted for. These systems have been developed for the past three decades and continue to evolve accompanying the technological progress. Given the complexity of the traffic flow models and the combinatorial problems that arise when trying to optimize the traffic operation, different simplifying approaches must be chosen in order to render the problem solvable in reasonable time. In such systems, the control actions are usually updated within two minutes, which demands great computational processing power.

The latest trend in traffic signal control research is the integrated use of real time information from individual vehicles. The technology behind it is called Car-to-Car and Car-to-Infrastructure Communications (C2X) and its use has been showcased in research projects like TRAVOLUTION[2] and KOLINE[3]. In this type of control system, the control entity benefits from more accurate information, like actual speeds and routes made available from the equipped vehicles, and also the vehicles themselves, which are informed about future signal switching times. With this information, drivers are able to reduce their speeds in order to avoid reaching queues before they dissipate; or increase them, within the speed limits, in order to catch the rest of the green light available.

Websites last visited on 10/09/2013

[1] TRL Limited, http://www.trlsoftware.co.uk/products/junction_signal_design/transyt

[2] http://www.travolution-ingolstadt.de/

[3] http://www.projekt-koline.de/

1.1 Objectives

Even with the just mentioned C2X technology being available for prototypes nowadays, the current work had its focus on improving an already existing ATCS without having to rely on future technologies whose adoption may still take another two decades to reach a reasonable market share. The penetration of such technology do not only depend on automobile manufacturers and traffic management system's fabricants to start equipping their products, but also depend on the standardization of the communication protocols, and the substitution of existing infrastructure by the public power.

The ATCS used as starting point for the current investigation is the Traffic-responsive Urban Control (TUC). After being involved in the development of the hardware and software for the implementation of this strategy for a pilot project in Brazil, as documented in Kraus et al. 2010, some potential improvement modifications were identified. TUC was conceived to be employed in arterial corridors where eventual routes crossing the main traffic artery have a much lower traffic demand. In such networks, the synchronization of the traffic lights belonging to the secondary crossing routes is not considered. But, for meshed networks, where the traffic demands are not much disparate, it may also be interesting and worthy to guarantee the maintenance of traffic lights synchronization for the secondary routes. Enabling this feature was the first goal of the present work. For the synchronization of traffic lights it is necessary to adjust the time difference between start of the green lights in the successive intersections. This time difference, called Offset, is dependent on the free flow speed of the route and the current queue length values. Since the evolution of queue sizes depend on traffic conditions, the offsets must be continuously adjusted throughout the day in order to guarantee a desirable operation. TUC's approach for the achievement of new offset values may be quite abrupt, leading to either too short green times or too long red times, which can unnecessarily harm traffic conditions. Trying to improve TUC's offset setting approach was the second objective.

Along with the two above mentioned goals, a new design for a Queue Estimator/ Predictor has been developed. Given TUC's traffic model requirements for current queue lengths, a new configuration of mathematical tools and traffic model is proposed for improving the queue estimation problem. The proposed method benefits from the consideration of the traffic dynamics in the whole network for achieving better estimation results. The technique requires only conventional traffic information gathered from ordinary inductance loop vehicle detectors.

1.2 Outline

In the following section, some of the terms and concepts that are basic in traffic signal control are reviewed.

In Chapter 2, an overview of the online urban traffic control systems currently available will be presented. It does not cover the entirety of them but gives a good idea of the existing systems and how they work.

Chapter 3 describes TUC in deeper detail. The original formulation is discussed, but more emphasis is given to the actual implementation used in this work.

After the description of TUC, the proposed modifications to the strategy are explained in Chapter 4. Furthermore, a much more straightforward counter proposal is offered, which will be used to evaluate the performance of the first one.

In Chapter 5 the queue estimator/predictor and its components are presented along with a concise overview on some of the other alternatives available.

The evaluation of all the developments of the current work is concentrated in Chapter 6. It is based on simulation tests performed in a microscopic traffic simulator over a real traffic network. The simulation comprises a whole day of traffic operation with traffic conditions similar to the real ones.

Finally, the conclusions are laid out in Chapter 7.

1.3 Basic Concepts and Terminology

Being the subject of this work, it is interesting to give a brief review on how traffic lights operate. In a given intersection, the traffic lights follow the so called *Signal Plan*, which aggregates all the information necessary for their operation. The Signal Plan encodes the duration of each color indication, their start time and the sequence that each traffic stream will be given the right-of-way (or simply, the green light). Since this process is repetitive, the Signal Plan only describes a single run of the successive switching of the traffic lights, which is denominated a Cycle. There may be more than one Signal Plan per intersection, which is individually followed according to the time of day, day of the week or special date.

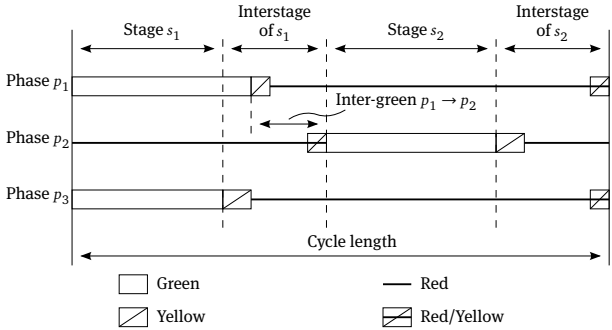


Figure 1.1: Cycle Progress

Each physical optical set, containing a green, a yellow and a red light is called a Signal Head, and the ones that operate in the exact same manner are grouped and referred to as a Signal Group. Usually, each Signal Group is associated to a single traffic stream. In the Signal Plan, the Signal Group is associated to a Phase, which describes the start time and duration of each color indication of a traffic light, beginning with the green. According to each case, it is possible to give right-of-way to more than one traffic stream at the same time. Whenever this is the intention, the Phases are grouped in a Stage. Since Phases may have different durations and start times, the encompassing Stage's green time has the start time of the latest starting Phase and its end coincides with the end of the earliest ending Phase. The yellow time of a given Phase is dependent on the free flow speed of the associated traffic stream, and should be long enough to avoid sudden brakings so that vehicles that are close to the stop line may keep on traveling and cross the intersection ahead. The red time of a given Phase only accounts the time interval necessary for the clearance of the intersection perimeter, assuring an all red interval among the conflicting traffic streams. The sum of the yellow time and the red time is called inter-green, which guarantees a safe change between conflicting Phases. The longest inter-green determines the boundaries of the Interstage, which is the time interval between two Stages. Depending on the order of occurrence of each Stage, i.e. the Phase switching order, the necessary red time must be accordingly adjusted. The sum of the duration of all Stages, and associated Interstages, is the Cycle length, as shown in Figure 1.1. The proportion between each of the green times available for the Stages is called Split.

With these basic elements of the Signal Plan presented, it is now possible to define the Offset. In a traffic network it is usually desirable that each pair of intersections operate in a synchronized way, enabling the creation of green-waves on determined routes, where the green lights are switched in a cascaded way so that the vehicles travelling through them are not forced to stop at every intersection. Each Signal Plan

has an Offset value that corresponds to the time interval between the beginning of the Cycle and a given time reference which may be a standard time of the day or the initial time instant where the Signal Plan starts to be valid. This offset value is the Absolute Offset and the difference between the Absolute Offsets of two consecutive intersections is the Relative Offset. The Relative Offset that describes the offset between the phases belonging to the first stage of each of the two involved signal plans will be referred to here as the Main Offset. Relative offset values that involve at least one stage that is not the first one in the Cycle will be referred to as Secondary Offset. This distinction is made to highlight one of the investigation points of the current work, where these offsets are explicitly taken into consideration during the calculation of the traffic control actions.

Since the different Signal Plans of a given intersection do not necessarily share the same Absolute Offset (i.e. depending on the traffic conditions and cycle length the desirable relative offset will vary), some sort of synchronization technique must be employed whenever a change in offset is necessary. As described in Shelby et al. 2006, there are many synchronization techniques available, also called offset transition techniques. They involve the implementation of successive modified signal plans until the desired absolute offset is reached. These modified signal plans have cycle lengths either shorter or longer than the original and current one, where the green durations change accordingly. Just to name a few of the transition methods, there is the *Dwell*; *Max Dwell*; *Add*; *Subtract*; *Shortway*; and *Immediate*. The reader may also refer to Pohlmann and Friedrich 2010 for a detailed description of them and other variations.

The last term that requires its definition for the current work is the saturation flow rate. It is a measure of the maximum amount of vehicles that are capable of crossing the stop line per unit of time, which is usually given in vehicles per hour. This magnitude is used in different traffic control techniques, mostly for the calculation of the desirable green time durations and offsets.

2 | Literature Review

The present overview of available Adaptive Traffic Control Systems (ATCS) does not intend to be a comprehensive review of them all, but gives a good insight on the existing alternatives.

2.1 Adaptive Traffic Control Systems

2.1.1 SCOOT

The Split, Cycle and Offset Optimization Technique (SCOOT) started its development in 1973 with the cooperation between the Transport and Road Research Laboratory (TRRL), the British Departments of Transport and Industry and the traffic systems companies: Ferranti, GEC and Plessey (Hunt et al. 1981), leading to the first full scale trial of the system in 1979 at the city of Glasgow (Hunt et al. 1982). The new system was developed to take advantage of the flourishing computer processing power, thus, offering a traffic control system capable of calculating signal plans online according to present traffic conditions. Since then, it has become one of the most widespread ATCS used worldwide.

The system operates in a centralized manner but the calculations do not consider the network effect of its actions. Each intersection is treated separately, even though it may consider the next downstream detector readings for correcting queue discharges. The traffic model used is based on the *Cyclic Flow Profile* (CFP) as depicted on Figure 2.1. There is one CFP for each monitored link of the network, which aggregates the occupancy and vehicle count information coming from the detector located upstream the stop line. Based on expected travel times and saturation flows it is capable of predicting the formation and dissipation of queues, which are used to decide about splits and offsets. Actually, SCOOT verifies its action alternatives based on a *Performance*

Index (PI). Three are the factors that add up to the PI: average predicted queues; number of stops; and congestion, measured according to the proportion of the cycle time that vehicles are stationary over the detectors. Each factor receives a weight before being add up to the PI, and this weight is the estimated additional cost incurred, e.g. $5 for each vehicle-hour of delay (ROBERTSON and BRETHERTON 1991). This index is also helpful for the traffic managers when evaluating traffic conditions. SCOOT also keeps track of the degree of saturation, i.e. the ratio of the average current flow and the maximum flow that each approach can handle, and is used to assess cycle times and green durations. The measured flows are also used to represent an overall "traffic demand" of the network, and comparing the evolution of the PI and the traffic demand is possible to identify accidents, since the PI increases and the demand remains practically the same in these cases.

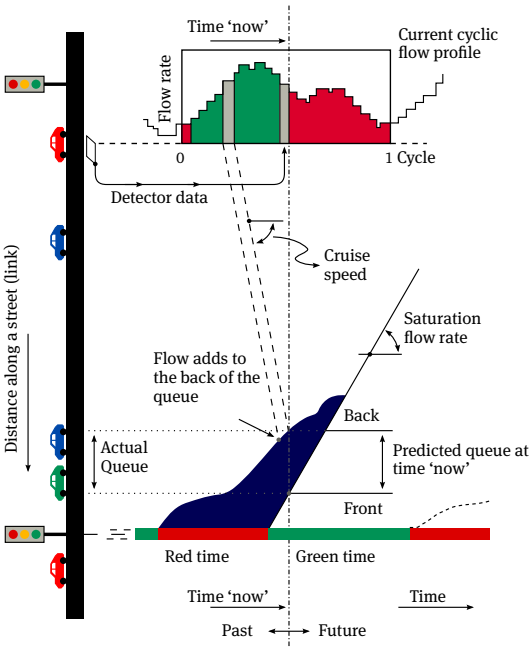


Figure 2.1: Principles of the SCOOT traffic model (source: SCOOT-UTC[1])

For each signal plan component (split, cycle and offset) there is one associated optimizer operating at its own pace as summarized in Table 2.1. Based on the current signal plan reference, the Split Optimizer decides whether the stage in course must

[1] *How SCOOT Works*: http://www.scoot-utc.com/DetailedHowSCOOTWorks.php?menu=Technical (visited on 10/09/2013)

be lengthened, shortened or neither. For each of these possibilities, and regarding only the associated intersection, the sum of the squared degrees of saturation of each link is compared and the alternative with the smallest sum is chosen (MING 1997). In case of modifications, the underlying stage length reference is also updated and used on the next cycle.

Optimizer	Frequency	Possible Changes [s]	Reference Changes [s]
Split	once for each stage change	0,±4	0,±1
Offset	at each new cycle	0,±4	0,±4
Cycle	once every 2.5 or 5 minutes	0,±4,±8,±16	0,±4,±8,±16

Table 2.1: Incremental SCOOT actions (source: MING 1997)

In a similar way, the Offset Optimizer analyses the upstream and downstream links of the junction that is about to end its cycle, and according to the PIs and CFPs generated as a result of each of the three possibilities (offset lengthening, shortening or neither), the one with the smallest sum of PIs is chosen. In case of traffic congestion occurring in short links, the Offset Optimizer prioritizes the maintenance of offsets on them even if this incurs into a larger PI sum.

The Cycle Optimizer chooses a common cycle for the network in which the degree of saturation of the most loaded junction retains a level of about 90%. In cases where the degree of saturation of any of the remaining junctions is sufficiently low, double cycling is used, i.e. the cycle of the given junction is set to the half of the common cycle (as long as it remains greater than/or equal to the minimum cycle allowed).

2.1.2 SCATS

The Sidney Coordinated Adaptive Traffic System (SCATS) was developed by the Roads and Maritime Services agency (former Department of Main Roads) of the New South Wales (NSW) state in Australia. Just like SCOOT, its first roll-out occurred in the late 70's and has since been adopted in many cities worldwide.

SCATS has no traffic model for calculating its control actions. All decisions are based on simple algorithms that incrementally adapt to traffic changes. Traffic conditions

are gathered from detectors positioned at the stop-line of the controlled intersections, where a higher correlation between signal timings and measurements holds (Sims and Dobinson 1980). The smallest control area is called a *sub-system*, and it may contain up to ten intersections where they share a common cycle length. According to special conditions, neighbouring *sub-systems* may be temporarily merged in order to obtain a possible gain in using a common cycle length, and therefore applying a desired offset between the adjacent intersections.

Operating in a cycle-by-cycle basis, the first measure of the system is to calculate the current *Degree of Saturation* (DS) of each of the green times in the *sub-system's* intersections. By analysing the occupancy profile of the detectors at the stop-line, the algorithm is capable of identifying the proportion of the green time that is efficiently used according to the available phase time, whose ratio is the above mentioned DS. The cycle length of the *sub-system* is then calculated as a function of the highest DS found. The changes in cycle length are limited to ±6s but may be increased up to +9s if the past two cycles were increased by +6s, letting the system react quicker to steep changes in demand.

For each intersection, SCATS holds a limited number of previously generated split plans that correspond to possible local traffic conditions. After the cycle length has been determined, the system evaluates each of these split plans, accordingly adjusted to the cycle length, and assigns a vote to the one that produces the expected DS for the critical approaches. After three consecutive votes, the split plan is finally implemented at the forthcoming cycle.

A similar scheme is used to determine the offsets. The algorithm evaluates a limited set of previously generated offset options and, at each new cycle, the offset plan that gets four consecutive votes is implemented. It is also possible to let this offset to be adjusted according to the current cycle length, either for queueing reasons or because of link speed changes provoked by heavy traffic. The offset may be shortened to increase residual queueing as demand arises; or may lengthened to accommodate longer vehicle platoons in heavy traffic conditions.

All the mentioned calculations take place in a centralized regional computer, at the so called strategic control level. When the signal plans are ready, they are transferred to the local traffic controllers, which then perform the tactical control. The information coming from detectors is used to change the phase lengths, where they may be: terminated earlier, when the demand is smaller than the average demand; omitted entirely, when demand actuated; or lengthened up until its maximum value, when existing demand requires it.

2.1.3 BALANCE

The Balancing Adaptive Network Control Method (BALANCE) was initially developed as part of two European research projects: Munich COMFORT (Csallner et al. 1995); and TABASCO (Catling and Harris 1995), at the Technische Universität München (Friedrich 1999). It has since been further developed and marketed by GEVAS[2] and TRANSVER[3] (Braun et al. 2008).

BALANCE has been projected as a three-tiered control architecture with robustness in mind. The premise was that it should be possible for one of the system's components to fail, or be inoperative, without compromising the operation of the whole system. At the top is the Strategic Control layer, where different set of goals for the long-term (with values aggregated in hourly figures) are laid out. The goals, like: travel time; delay; number of stops; emissions; etc., are set for each route or sub-network of choice and later combined in a Performance Index, by associating a weight to each. This PI is then passed to the next control layer, the Tactical Control, to be used as guideline for its calculations.

The Tactical Control may harbor different control strategies, be it for highways or urban roads, aiming the mid-term control, i.e. the next 5–15 minutes. It uses an internal traffic model to evaluate an explicit objective function that conforms to the goals received from the Strategical Control layer. Before any optimization takes place, the current state of the network is estimated. For obtaining traffic demands and turning rates on each signal group, it uses the method of entropy maximization proposed in van Zuylen and Willumsen 1980, and for estimating queues and delays it offers the choice of two different methods: one proposed in Kimber and Hollis 1979; and another using Markov-chains. Once the network's current state is estimated, the critical intersection is used to determine the cycle length that minimizes delay. Based on pre-determined and possible stage sequences, the algorithm calculates thresholds for each one of the stages resulting in a frame signal plan that is transmitted to the Local Control. Offsets are calculated individually for each pair of intersections, starting from the critical ones. All possible offset alternatives are tested and the one resulting in the lowest traffic delay is chosen. In its latest developments, the GALOP-Online method (Braun and Kemper 2008) was introduced in substitution of the original hill-climbing algorithm used for the optimization (Braun et al. 2008). It is a genetic algorithm that searches for the best combination of cycle length, offsets and phase sequences comprising all intersections, and that minimizes the given objective function.

The last step of the control is performed locally, at the traffic controller device. Based on the frame signal plan processed by the Tactical Control, the phase lengths are con-

[2] GEVAS Software GmbH, http://www.gevas.eu/ (visited on 10/09/2013)
[3] TRANSVER GmbH, http://transver.de/ (visited on 10/09/2013)

stantly adapted to the small dynamical changes in traffic that are captured through the traffic detectors. The actuated traffic control method used may be, for example, the VS-PLUS from VS[4] or TRENDS from GEVAS, which also include public transport prioritization.

2.1.4 OPAC

The Optimization Policies for Adaptive Control (OPAC) was originally developed at the University of Lowell, Massachusetts (Gartner 1982, 1983; Gartner et al. 1983). Its last iteration has received support from the American agency: FHWA (Federal Highway Administration), and the PB Farradyne Inc., where it was included in the RT-TRACS (Real-Time Traffic Adaptive Signal Control System) project as one of the alternative online traffic control strategies (Gartner et al. 2001; Pooran et al. 1996).

OPAC, at its debut, was presented as a suite of three alternatives that actually represents the progress of its development. The first one, OPAC-I, lays the foundations of the technique where the traffic problem is solved with a Dynamic Programming (DP) method — a global optimization strategy for multi-stage processes that outputs global optimal solutions (Gartner et al. 2001). Since it demands full availability of traffic states for the period of the optimization, and demands a great deal of processing time, it could not be used for online traffic control and remained as a reference case for evaluating the relative effectiveness of other online-suited alternatives.

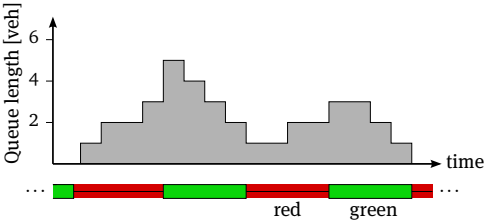


Figure 2.2: Hypothetical queue profile in OPAC

By using DP, the traffic control problem, limited to a given time span, may be divided into N sub-problems, in this case called stages, which can be solved individually, thus, much more computationally efficient. Each sub-problem/stage is characterized by a discrete time interval of 2–5s, during which it must decide whether or not to end a given green phase, i.e. start or not the other possible ones. Each intersection is treated individually and there is no concept of cycle, phase order or offset involved,

[4] VS – Verkehs-Systeme AG, http://www.vs-plus.com/ (visited on 10/09/2013)

just maximum and minimum values are enforced. The goal of the optimization is to reduce the vehicle delays during the specified control horizon, which is measured as the area under the queue-length curve of each approach, as shown in Figure 2.2. The solution of the problem is carried out backwards, i.e. it solves the last sub-problem at the end of the control horizon and starts coming back until the first one.

Pursuing the same path, OPAC-II constitutes a simplification of the first one enabling it to be used online. The sub-problem now consists of a 50–100s time frame where at least one, and utmost three phase changes are enforced (Gartner 1984). Through the use of a sequential constrained search, the objective function is evaluated sequentially for all feasible switching sequences. The current objective value is compared to the one previously stored, and in case of being smaller, it substitutes the previous one. This optimization is carried out independently for each sub-problem, and, unlike OPAC-I, it progresses forward in time. Comparisons against OPAC-I have shown that this second alternative represents a degradation of around 10% in performance.

Similar to OPAC-I, OPAC-II also demands the availability of vehicle arrival for the whole control horizon, which cannot be precisely predicted. In order to circumvent this shortcoming, OPAC-III was proposed. It employs a rolling horizon framework, which divides the stage in n intervals and uses fresh arrival data for the first intervals of the stage and an updated average for the rest of the stage. Since only the optimization results for the first intervals are used, which are fed with actual arrival data, the successive optimization rounds, that occur for each stage, are capable of delivering better results.

For the RT-TRACS project, OPAC's control logic was extended to include a coordination strategy, so that it would be suitable for use in arterials and networks (Gartner et al. 2001). Called Virtual-Fixed-Cycle OPAC (VFC-OPAC), the extension implements a three level control architecture. The first level, the Local Control Layer, is the known OPAC-III; the second level, the Coordination Layer, calculates, once per cycle, desirable offsets for each intersection; and the third level, the Synchronization Layer, enforces the cycles for each group of intersections that should be synchronized.

2.1.5 RHODES

The Real-Time Hierarchical Optimized Distributed Effective System (RHODES) has been conceived at the University of Arizona by the Advanced Traffic and Logistics Algorithms and Systems (ATLAS) research center (Mirchandani and F.-Y. Wang 2005).

The system is divided in three hierarchical control levels, as proposed in Head et al. 1992, covering different network resolution degrees, as shown in Figure 2.3. The system collects information from the street detectors and performs a prediction of future

traffic streams at various degrees of aggregation, depending on which control level it will be used. The highest level has a "dynamic network loading" model that reflects the slow-varying characteristics of traffic. It encompasses network geometry with available routes, with the associated Origin/Destination matrices, and possible road detours caused by construction sites. At this level, coarse hourly traffic load estimates are calculated and passed to the next control level: *Network Flow Control*. At the middle level, these estimates are further processed into demand patterns, which reflect the sizes of vehicle platoons and their speeds. The information about the platoons is used to allocate the green times. These preliminary green times are then fine tuned at the lower level, the *Intersection Control*, where the phase changes are settled according to observed and predicted arrivals of individual vehicles.

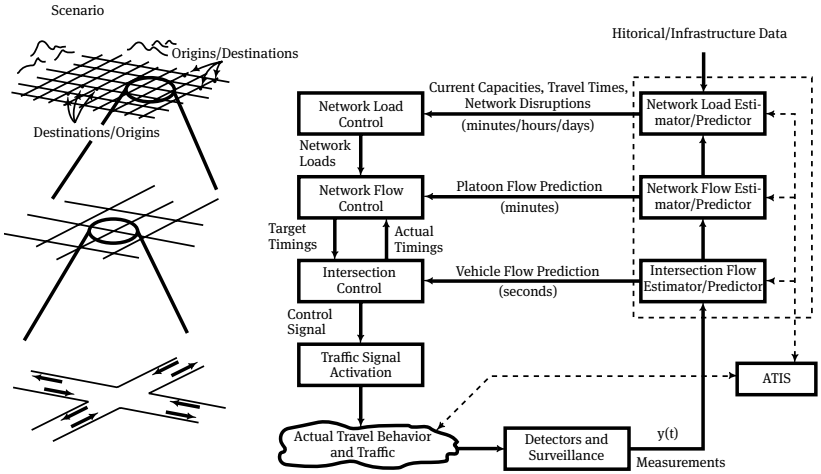


Figure 2.3: RHODES architecture (source: MIRCHANDANI and HEAD 2001)

At the *Network Flow Control*, the information about the moving platoons in the network, produced by the APRES-NET model (DELL'OLMO and MIRCHANDANI 1996), is used to determine the desired offsets between the intersections. The APRES-NET model is a simplified traffic simulation model capable of propagating platoons of vehicles through a sub-network of intersections. It is not only used to generate the platoon information but also to evaluate the different possible offset configurations created with the REALBAND algorithm (DELL'OLMO and MIRCHANDANI 1995). Given platoon sizes, positions and speeds in a subset of neighbouring links, the REALBAND algorithm builds a binary decision tree that depicts, in a pre-defined horizon of about 200–300 seconds, the possible conflicts between platoons trying to cross the intersections. The tree nodes represent three possible decisions: allow platoon A to pass without stopping; allow platoon B to pass without stopping; or split platoon A (or B).

Each scenario is evaluated using the APRES-NET model, and, according to the specified performance index (stops and/or delay) the best alternative is chosen.

At the next control level, the *Intersection Control* level, the PREDICT algorithm (Head 1995) uses detector data along with planned phase timings from upstream intersections to predict future vehicle arrivals on the downstream ones. It assumes that the arrival process can be divided into a predictable and an unpredictable factor. According to their relative proportion, the control strategy may choose to enforce platoon progression (higher predictable factor), or to gather the arrivals into platoons in the downstream intersections (higher unpredictable factor). For the algorithm to work, the information from two detectors is necessary, one located close to the link's entrance and the other close to the stop line. This information is used to calculate current estimates for link travel times, turning probabilities, queue sizes and queue discharging rates. After the prediction step, the control strategy, based on dynamic programming (Bellman 1954), optimizes the phase lengths in a rolling horizon framework (with one second resolution), where phases may be repeated or completely left out. Whenever there is new data, the current phase may be interrupted or extended, given the new results of the optimization. The optimization objective may be set to minimize average delays or number of stops, or even maximize throughput (Sen and Head 1997), and it is performed under some desired constraints like minimum greens, phase coordination and, when required, phase sequence.

2.1.6 Pohlmann's

The ATCS presented in Pohlmann 2010 has been developed as a result of a research project, sponsored by the German Research Foundation (DFG - *Deutsche Forschungsgemeinschaft*), at the Institute for Transportation and Urban Engineering (IVS - *Institut für Verkehr und Stadtbauwesen*, *Technische Universität Braunschweig*).

The system comprises two main modules: the Traffic State Estimator/Predictor; and the Traffic Signal Plan Optimizer. Unlike the other systems presented here, the concept of Pohlmann's ATCS has a much more conservative approach. Instead of trying to respond to traffic changes on a second-by-second or cycle-by-cycle basis, Pohlmann's solution performs an update to the resulting signal plans every 15 minutes, as shown in Figure 2.4.

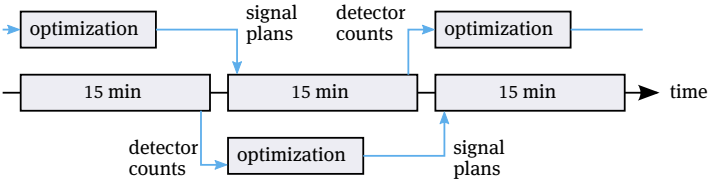


Figure 2.4: Update process (source: POHLMANN 2010)

It is possible to point out two reasons for this choice of the update period's length. One is related to the necessary time the optimization technique employed takes to perform its calculations. The other is the author's own vision on how often it is really necessary to apply modifications to the running signal plans. The resulting effect is a more robust control system, but also a slower responding one to adverse incidents.

Given the update scheme presented, it is necessary to produce control measures that will be compatible with the active traffic conditions in one time period ahead, i.e. 15 minutes. In order to generate reliable results, the Traffic State Estimator/Predictor heavily bases its predictions on stored historical measurements. This approach, based in the work presented in FÖRSTER 2008, combines current 15-minute detector counts with four previous ones, forming a traffic pattern of one hour duration. This traffic profile is compared to different other traffic profiles stored in the database, which have been gathered for a reasonable amount of time. These traffic patterns are formed by the concatenation of consecutive 15-minute averaged traffic counts summing up to a whole day of measurements. There may be different traffic profile groups stored, e.g. for different days of the week and/or time of the year. At each new day, the collected traffic profile is asserted to the existing groups, and added to the corresponding one, therefore updating the average. The algorithm searches for the most similar sub-pattern (1 hour long) in the database, in shape and magnitude, belonging to a whole day pattern (24 hour long) group. From the found matching sub-pattern it looks ahead in time, precisely two 15-minute blocks ahead in the respective whole pattern, and uses the second consecutive 15-minute average as basis for the calculations of the next signal plans.

After the averaged traffic data has been selected from the database, it is further processed to generate expected turning rates, i.e. routing information, and an integral estimation of the traffic volumes present in all links of the network, including the ones not equipped with traffic sensors. This is accomplished by the use of a technique called Information Minimization (IM), which was introduced by VAN ZUYLEN and WILLUMSEN 1980 and further developed by Pohlmann himself, following the previous improvements in FRIEDRICH and Y.-P. WANG 2006, 2008; Y.-P. WANG 2008. IM

is a traffic assignment technique originally designed to estimate OD matrices, but in this case it is also employed to estimate the traffic volumes in the network, which will be necessary for the Traffic Signal Plan Optimizer. The only problem of this approach is that it is only applicable in under-saturated to saturated scenarios, turning the system as a whole unusable for the traffic congested cases.

Once all expected traffic volumes and turning rate information is available, the Traffic Signal Plan Optimizer module is called. Its core is based on the Cell Transmission Model (CTM) (Daganzo 1994), a first order macroscopic traffic flow model capable of reproducing all traffic conditions and characteristics (with the exception of platoon dispersion) with a good degree of fidelity. The traffic volumes and turning rates found are directly applied to the corresponding structures in the CTM and considered constant for the whole simulation period. Given the complexity of the CTM, and therefore the time required for computing optimal traffic signal plans with it, it is only employed for the calculation of the offsets, leaving the calculation of common cycle and splits to a more straightforward approach. Two alternatives are presented for setting the common cycle, both advised by HBS 2009 and RiLSA 2010, the German Traffic Manual and Guidelines, respectively. One of them is the renowned Webster's Formula (Webster 1958), and the other called Saturation Based Cycle, which outputs the minimum cycle needed for serving the traffic demand in utmost one cycle period. These calculations are performed for each individual intersection, and the biggest cycle found is used as common cycle. The splits are proportionally distributed according to each phase demand and constrained by the minimum green times. With these values fixed, different offset configurations are evaluated through the simulation of the CTM for the same 15-minute period. For each simulation run of the CTM, a different combination of offsets is tested. Given the combinatorial explosion of the number of possible offset alternatives, the author uses a Genetic Algorithm, following the works in Almasri 2006, to find the best offset scheme. For each evaluation performed, the necessary transition cycles are also included.

2.1.7 TUC

Traffic-responsive Urban Control (TUC) is a technique initially developed as part of an integrated traffic control system for corridor networks in the context of the European project TABASCO (Transport Applications in Bavaria, Scotland and Others) (Catling and Harris 1995; Papageorgiou 1995). It was conceived to deal with saturated traffic conditions with the aim of reducing the risk of queue spillbacks and oversaturation. In the first version (Diakaki et al. 2002), it was only capable of determining the green times of each stage given predefined values of cycle length and stage order. In a subsequent iteration (Diakaki et al. 2003), three additional modules were added: Cycle

Control; Offset Control; and Public Transport Priority, that along with the prior Green Control Module turned it into a more complete urban traffic control solution.

TUC's approach to traffic control relies on the use of straightforward strategies to calculate the control actions that will make the traffic operate at desirable levels. By using a simple, yet effective, macroscopic traffic flow model, the Green Control Module is capable of taking into consideration the whole network, almost independent of its size, in the calculation of the length of each green time. This has an important advantage because the operation of a given intersection interferes with the surrounding ones, and this interdependence is being considered. Since the technique was envisaged with corridor networks in mind, the Offset Control Module implements the offsets regarding only the corridors. Therefore, it just consists of a simple maintenance of green waves in these main routes, so that standing queues are dissipated before the platoons from upstream intersections arrive in the downstream ones.

Since TUC is used as base for the current investigation, a better and more detailed description of its concept will be presented on Chapter 3.

3 | TUC

As mentioned on the previous Chapter, TUC (Traffic-responsive Urban Control) was used as the starting point of the current work. Actually, the original implementation, as in Diakaki et al. 2003, 2002, was not followed completely and the TUC strategy that will be presented here is more faithful to its further development, as published in Aboudolas et al. 2009, 2010. Basically, the main differences from the original implementation lie on the control/optimization technique used and the way the common cycle of the network and the offsets are calculated. The concept and structure of the strategy remains the same.

3.1 Control Scheme

TUC is divided in four modules: Cycle Control; Offset Control; Green Control; and Public Transport Priority. They are hierarchically dependent and each one of them has its own update cycle. The Cycle Control Module dictates the common cycle length of the network according to its traffic conditions. The common cycle is then passed to the Offset Module which is in charge of not only defining the offsets between each pair of intersections, but also calculating the transition cycles needed in order to achieve these offsets. Whenever the Offset Cycle calculates new offsets for the network, there will usually be different cycle lengths, the transition cycles, that each intersection must implement. These cycles are passed as parameter to the Green Control Module responsible for distributing the green times of each stage of the cycle. In case there is no change in offsets, the current common cycle is used by the Green Control Module. The Public Transport Priority Module is not object of the current work and therefore will not be presented, but it suffices to state that this module acts directly in the Green Control Module by giving more green to the phase serving the public transport vehicle during its drive through of the according link.

TUC is a very straightforward traffic control solution, which may be applied to very large networks. Nevertheless, the traffic engineer may decide to divide the network in sub-networks, operating independently from one another, like presented in KRAUS et al. 2010. But, whenever the physical delimitation of each sub-network has been fixed, it is not possible to dynamically change the control structure, as foreseen in SCATS for example.

3.2 Cycle Control Module

The Cycle Control Module in DIAKAKI et al. 2003 uses a simple feedback algorithm, known as Proportional Controller (OGATA 2010), to adjust the common cycle length C^{ref} of the network, which can also be called cycle reference.

A pre-specified percentage of the network links is used to calculate an average load $\bar{\lambda}$ of the network. The load λ_z of a given link z, is considered to be the ratio of the current amount of vehicles x_z and the maximum number of vehicles allowed $x_z^{\max}$:

$$\lambda_z = \frac{x_z}{x_z^{\max}} \tag{3.1}$$

The chosen links are the ones currently with the highest loads. This averaged load is then compared to an expected nominal load λ^{nom}, and the difference between the two, multiplied by the control parameter K^{cont}, is used to either increase or decrease the value of a predefined nominal cycle C^{nom}:

$$C^{\text{ref}} = C^{\text{nom}} + K^{\text{cont}}\left(\bar{\lambda} - \lambda^{\text{nom}}\right) \tag{3.2}$$

For the intersections with sufficiently low saturation levels, a double-cycling feature is proposed, where the cycle applied in these cases is half of the one calculated for the network.

Since the focus of the current investigation lies on the Offset Module of the TUC strategy, it has been decided that it would be interesting to keep some similarities to the other ATCS (POHLMANN 2010) used as reference in the evaluation of the current proposal. With this in mind, this simple feedback algorithm was substituted with the same type of cycle adjustment used in Pohlmann's work, which also means that the double-cycling feature was left out. Actually there were two alternatives in Pohlmann's ATCS, one making use of minimum delay cycle formula from WEBSTER 1958, and the other taken from HBS 2009 or RILSA 2010, called Saturation Based Cycle. This was done in order to reduce the differences between the two solutions and to be able to better understand possible discrepancies.

3.2.1 Webster's Cycle Alternative

Webster has derived a formula for the calculation of cycle lengths based on a series of experiments on isolated intersections. His method minimizes the mean overall delay of vehicles being served by the controlled intersection in question. The distribution of green times for each phase is done according to the ratio between the phase's demand and the sum of all relevant demands. Even though his findings were actually only valid for isolated intersections, where the arrival of vehicles is considered random, its simple applicability gained great popularity and has been widely used.

The utilization of the Webster's method in the current work has followed just about the same procedure as in Pohlmann 2010. The difference lies in the fact that the amount of green time distributed to the phases is not determined by Webster's method, but are calculated by the Green Control Module (Section 3.4).

At each update time, the Webster cycle length of each intersection in the network is calculated:

$$C_{\mathcal{J}_n}^{\mathrm{WEB}} = \frac{5 + 1.5 \sum_{i \in \mathcal{S}_{\mathcal{J}_n}} L_i}{1 - \sum_{j \in \mathcal{S}_{\mathcal{J}_n}} \frac{q_j^{\mathrm{crit}}}{q_j^{\mathrm{sat}}}} \tag{3.3}$$

where:

- $\mathcal{J}$ set containing all junctions
- $\mathcal{J}_n$ junction n
- $C_{\mathcal{J}_n}^{\mathrm{WEB}}$ Webster's advised cycle length for junction n, [s]
- $\mathcal{S}_{\mathcal{J}_n}$ set containing all stages of junction n
- L_i interstage of stage i, [s]
- q_j^{crit} traffic demand on the critical phase of stage j, [veh/h]
- q_j^{sat} saturation flow of the link being served by the phase related to q_j^{crit}, [veh/h]

The critical phase of a given stage is the one with the highest demand. For the cases where a phase spans over more than one stage the sum in the denominator of Equation 3.3 may have to be altered. Take, for example, a hypothetical signal plan with the stage sequence shown in Figure 3.1, where p_i is the phase and s_j the stage.

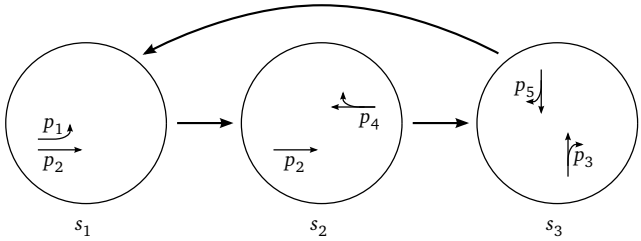


Figure 3.1: Multiple-stage phase

Throughout the text p and s will be used either to identify the phase/stage in question, or the length of the associated green time available to each one of them. Note that phase p_2 belongs to stages s_1 and s_2. If the traffic demand for p_2, q_{p_2}, is bigger than the sum of the traffic demands q_{p_1} and q_{p_4}, i.e. $q_{p_2} \geqslant q_{p_1} + q_{p_4}$, then the critical phase of both s_1 and s_2 would be p_2. In this case, one of the stages must be disconsidered so that q_{p_2} is accounted only once in the sum appearing in the denominator of Equation 3.3. But, if $q_{p_2} < q_{p_1} + q_{p_4}$, then the critical phase of s_1 becomes p_1, and the critical phase of s_2 becomes p_4, even if $q_{p_2} > q_{p_1}$ or $q_{p_2} > q_{p_4}$.

Another exception, that must be mentioned, happens when a given stage s_j has a fixed length, which is the usual case for pedestrian lights. In this case, Equation 3.3 must also be altered to incorporate the fixed length stages:

$$C_{\mathcal{J}_n}^{\text{WEB}} = \frac{5 + 1.5 \sum_{i \in \mathcal{S}_{\mathcal{J}_n}} L_i + \sum_{j \in \mathcal{S}_{\mathcal{J}_n}^{\text{fix}}} s_j}{1 - \sum_{j \in \mathcal{S}'_{\mathcal{J}_n}} \frac{q_j^{\text{crit}}}{q_j^{\text{sat}}}} \tag{3.4}$$

where:

$\mathcal{S}_{\mathcal{J}_n}^{\text{fix}}$ set containing all fixed length stages of junction n

$\mathcal{S}'_{\mathcal{J}_n}$ set containing all stages of junction n that have no fixed length

After each junction's cycle have been calculated with Equation 3.3, the biggest cycle length found is used as the common cycle of the network:

$$C^{\text{ref}} = \max\{C_{\mathcal{J}_n}^{\text{WEB}}, \forall n \in \mathcal{J}\} \tag{3.5}$$

3.2.2 Saturation Based Cycle Alternative

The HBS, the German Handbook for the Dimensioning of Road Infrastructure, presents, along with the Webster's method, a similar formula for the calculation of cycle lengths. It states that this second alternative should be employed whenever the implementation of green waves is desired.

The Saturation Based Cycle is derived by assuming that the traffic demand of a determined phase should be served at most in one green time period. This means that the same amount of vehicles that arrived past the end of the last green, must be able to pass on the next green:

$$q_{p_i} C = q_{p_i}^{\text{sat}} p_i \tag{3.6}$$

$$\Rightarrow p_i = \frac{q_{p_i} C}{q_{p_i}^{\text{sat}}} \tag{3.7}$$

Accounting only for the critical phases in each stage, it is possible to derive the needed cycle length for $\mathcal{J}_n$:

$$\begin{aligned} C_{\mathcal{J}_n} &= \sum_{j \in \mathcal{S}_{\mathcal{J}_n}} \frac{q_j^{\text{crit}} C_{\mathcal{J}_n}}{q_j^{\text{sat}}} + \sum_{i \in \mathcal{S}_{\mathcal{J}_n}} L_i \\ &= C_{\mathcal{J}_n} \sum_{j \in \mathcal{S}_{\mathcal{J}_n}} \frac{q_j^{\text{crit}}}{q_j^{\text{sat}}} + \sum_{i \in \mathcal{S}_{\mathcal{J}_n}} L_i \end{aligned} \tag{3.8}$$

$$\Rightarrow C_{\mathcal{J}_n} \left(1 - \sum_{j \in \mathcal{S}_{\mathcal{J}_n}} \frac{q_j^{\text{crit}}}{q_j^{\text{sat}}} \right) = \sum_{i \in \mathcal{S}_{\mathcal{J}_n}} L_i$$

$$\Rightarrow C_{\mathcal{J}_n} = \frac{\sum_{i \in \mathcal{S}_{\mathcal{J}_n}} L_i}{1 - \sum_{j \in \mathcal{S}_{\mathcal{J}_n}} \frac{q_j^{\text{crit}}}{q_j^{\text{sat}}}} \tag{3.9}$$

Considering the stochasticity of the real life vehicle arrival behaviour, the factor c_j is introduced in Equation 3.9. This term, called degree of saturation, reduces the expected saturation flow of the associated critical traffic demand being served in stage

j and confers the final form of the Saturation Based Cycle $C^{\text{SAT}}_{\mathcal{J}_n}$ equation:

$$C^{\text{SAT}}_{\mathcal{J}_n} = \frac{\sum_{i \in \mathcal{S}_{\mathcal{J}_n}} L_i}{1 - \sum_{j \in \mathcal{S}_{\mathcal{J}_n}} \frac{q^{\text{crit}}_j}{c_j q^{\text{sat}}_j}} \tag{3.10}$$

According to RILSA 2010, the degree of saturation c_j should be set with a value in the range $[0.8, 0.9]$, and at the present work a value of 0.85 has been chosen. For the special cases presented in Section 3.2.1, like the multiple-stage phase and the fixed length stage, Equation 3.10 must also be updated and becomes:

$$C^{\text{SAT}}_{\mathcal{J}_n} = \frac{\sum_{i \in \mathcal{S}_{\mathcal{J}_n}} L_i + \sum_{j \in \mathcal{S}^{\text{fix}}_{\mathcal{J}_n}} s_j}{1 - \sum_{j \in \mathcal{S}'_{\mathcal{J}_n}} \frac{q^{\text{crit}}_j}{c_j q^{\text{sat}}_j}} \tag{3.11}$$

At last, the biggest cycle length found is used as the common cycle of the network:

$$C^{\text{ref}} = \max\{C^{\text{SAT}}_{\mathcal{J}_n}, \forall n \in \mathcal{J}\} \tag{3.12}$$

3.3 Offset Control Module

The Offset Control Module was conceived with traffic corridors in mind. In this type of network the traffic demands in the main road/arterial are much larger compared to the secondary crossing roads. Therefore, the latter may be overlooked with respect to the synchronization between their consecutive intersections. The goal of this module is the maintenance of "green waves" in the main roads. The term comes from the visual effect perceived by the drivers in which the traffic lights in the successive intersections turn green just in time to let them drive through them without stopping. Green waves, in traffic corridors, are not only expected from the drivers, but also beneficial in reducing delays, fuel consumption and nocive emissions.

As common practice, the stage that is assigned to give right of way to the main road is the first one of a signal plan. Since it starts with the beginning of every cycle, it is much easier for traffic planners to adjust the offsets, because this parameter, in a traffic controller device, is measured according to the beginning of the cycle.

Based on the fact that the Offset Control Module can only interfere with the length of every cycle, because the duration of each stage is determined later by the Green Control Module, it is only capable of maintaining control of the relative offsets between the main/first stages of consecutive intersections.

In order to calculate the necessary offsets, capable of guaranteeing the formation of a green wave, not only the distances between each consecutive intersection and the nominal travel speeds are needed, but also an estimate of the queue lengths of the downstream intersections must be considered. Following the works of ABU-LEBDEH and BENEKOHAL 1997, the offset is based on the calculation of the time needed for the first vehicle, coming from the upstream intersection, to meet the end of the downstream dissipating queue just when the last vehicle in the queue starts moving. Assuming two consecutive intersections A and B connected by the link z, with length l_z^{link} and nominal free-flow speed v_z^{free}, it is possible to approximate the queue length, with the use of Equation 3.1, to $\lambda_z l_z^{\text{link}}$. And the time for a vehicle coming from the upstream intersection A to reach the end of the queue is:

$$\frac{l_z^{\text{link}} - \lambda_z l_z^{\text{link}}}{v_z^{\text{free}}} \Rightarrow \frac{(1-\lambda_z) l_z^{\text{link}}}{v_z^{\text{free}}} \tag{3.13}$$

The kinematic wave originated from the switching of the green light in the downstream intersection B travels down the queue with speed v^{bw} (considered equal to 15 km/h in DIAKAKI et al. 2003), and the time to reach the end of the queue is:

$$\frac{\lambda_z l_z^{\text{link}}}{|v^{\text{bw}}|} \tag{3.14}$$

Equation 3.13 represents the time the green light of intersection B will have to wait to start, so that the vehicles coming from A have enough time to reach B, a positive value. Equation 3.14 represents the time the green light of intersection B will have to start before the green light on A, so that there is enough time to dissipate the queue, a negative value. By adding both values, the offset between A and B, $\phi_{A,B}^{\text{ref}}$ is then given:

$$\begin{aligned} \phi_{A,B}^{\text{ref}} &= \frac{(1-\lambda_z) l_z^{\text{link}}}{v_z^{\text{free}}} - \frac{\lambda_z l_z^{\text{link}}}{|v^{\text{bw}}|} \\ &= \frac{l_z^{\text{link}}}{v_z^{\text{free}}} - \frac{\lambda_z l_z^{\text{link}}}{v_z^{\text{free}}} - \frac{\lambda_z l_z^{\text{link}}}{|v^{\text{bw}}|} \\ &= \frac{l_z^{\text{link}}}{v_z^{\text{free}}} - \lambda_z l_z^{\text{link}} \frac{|v^{\text{bw}}| + v_z^{\text{free}}}{v_z^{\text{free}} |v^{\text{bw}}|} \end{aligned} \tag{3.15}$$

$$\Rightarrow \phi_{A,B}^{\text{ref}} = \frac{l_z^{\text{link}}}{v_z^{\text{free}}} - \frac{x_z}{x_z^{\max}} l_z^{\text{link}} \frac{|v^{\text{bw}}| + v_z^{\text{free}}}{v_z^{\text{free}} |v^{\text{bw}}|} \tag{3.16}$$

A slightly different approach has been used throughout the current work. Instead of considering the meeting point: the end of the queue, it has been used the stop line of intersection B. By doing this, it is possible to use the local saturation flow as a measure of how fast the queue dissipates:

$$\phi_{A,B}^{\text{ref}} = \frac{l_z^{\text{link}}}{v_z^{\text{free}}} - \frac{x_z}{q_z^{\text{sat}}} \tag{3.17}$$

so that the platoon coming from A will reach the last vehicle that stood in queue when it crosses the stop line of B. Using Equation 3.17 instead of 3.16 has the advantage of allowing the use of the parameter q_z^{sat}, which may be interesting to be fine tuned since it is an important parameter in the traffic flow model used in the Green Control Model.

The Offset Control Module foresees the onset of offsets in both directions of a two-way road, as long as they are between the main/first stages of each intersection. But only the direction with the most demand gets its offset implemented. Note that in a hypothetical case, where there is no demand in both directions, the signs of the offset values in each direction would be inverted. The offsets between each pair of intersections are calculated independently from one another, but in a successive manner.

Diakaki et al. 2003 mention the cases where there may be intersecting arterials, and where a priority order is specified so that the offsets of the arterials with highest demands are first implemented. But, they are not clear how this would be possible because this case implies that the crossing, with less priority, arterial would have to be given a secondary stage as its right of way giving green time. Meaning that the implementation of the secondary offset, that shares the intersection with the main offset, would not be feasible anyway.

After calculating the desirable offsets of all pairs of intersections on the arterials, the Module calculates the one cycle transition plan for each intersection that will bring them to the aimed offsets. The technique used is similar to the *Immediate* method (Lee and Williams 2009), where the offset is achieved in one single transition cycle. The difference is that all stages may be extended or shortened as would be the case in the *Shortway* method. The transition cycle is limited to a maximum and minimum value ($C^{\max}$ and $C^{\min}$), and is successively calculated for the downstream intersections of the arterial.

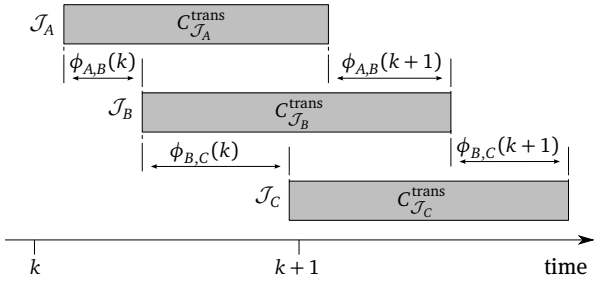


Figure 3.2: Transition cycle

Just for illustration, as shown in Figure 3.2, imagine a hypothetical arterial with the following set of consecutive junctions $\mathcal{J} = \{A, B, C\}$ and associated set of offset pairs: $\mathcal{O} = \{(A, B), (B, C)\}$, regardless of whether it is a two-way road or not. The figure shows the progression in time of the cycles of the mentioned intersections during the future time interval $k+1$, and the resulting offsets at its end. As first step, the algorithm makes the transition cycle $C^{\text{trans}}_{\mathcal{J}_A}$ equal to the common cycle C^{ref}, then it calculates the necessary $C^{\text{trans}}_{\mathcal{J}_B}$ so that the new $\phi_{A,B}(k+1)$ can be reached, taking into consideration the former offset between the two $\phi_{A,B}(k)$:

$$\phi_{A,B}(k+1) = \phi_{A,B}(k) + C^{\text{trans}}_{\mathcal{J}_B} - C^{\text{trans}}_{\mathcal{J}_A} \tag{3.18}$$

The algorithm then moves to the next offset $\phi_{B,C}$. In possession of the new $C^{\text{trans}}_{\mathcal{J}_B}$, the downstream $C^{\text{trans}}_{\mathcal{J}_C}$ is calculated.

Algorithm 3.1: Transition Cycle Filter

mod(x, y) : Function that returns the remainder of the division of x by y;
foreach $n \in \mathcal{J}$ **do**
  if mod$\left(C^{\text{trans}}_{\mathcal{J}_n}, C^{\text{ref}}\right) \geqslant C^{\min}$ **then**
    $C^{\text{trans}}_{\mathcal{J}_n} \leftarrow$ **mod**$\left(C^{\text{trans}}_{\mathcal{J}_n}, C^{\text{ref}}\right)$;
  else
    $C^{\text{trans}}_{\mathcal{J}_n} \leftarrow$ **mod**$\left(C^{\text{trans}}_{\mathcal{J}_n}, C^{\text{ref}}\right) + C^{\text{ref}}$;
  if $C^{\text{trans}}_{\mathcal{J}_n} > C^{\max}$ **then**
    $C^{\text{trans}}_{\mathcal{J}_n} \leftarrow C^{\max}$;

At last, the calculated transition cycles are formatted to be in the range of the allowed maximum and minimum cycles in a further filtering step, depicted in Algorithm 3.1 Transition Cycle Filter. In the cases where $C^{\text{trans}}_{\mathcal{J}_n}$ is limited by $C^{\max}$, the implemented

offset remains with an error, of magnitude $C_{\mathcal{J}_n}^{\text{trans}} - C^{\max}$, until new offsets are calculated by the Offset Control Module.

3.4 Green Control Module

The Green Control Module is the last one to be invoked by the TUC strategy. It will use either the common cycle length determined by the Cycle Control Module, or the transition cycle required by the Offset Control Module, when it is available.

The Module uses a traffic flow model that depicts the dynamics of the entire network in order to decide the amount of green time needed by each phase.

3.4.1 Traffic Flow Model

The traffic flow model used is a macroscopic model based on the findings of Gazis and Potts 1963, and called the store-and-forward model. It consists of a linear relationship between the number of vehicles present in one link, the green time and the traffic demand during one cycle period. Its simplicity allows the employment of efficient algorithms for solving the control problem of urban traffic networks (Aboudolas et al. 2009).

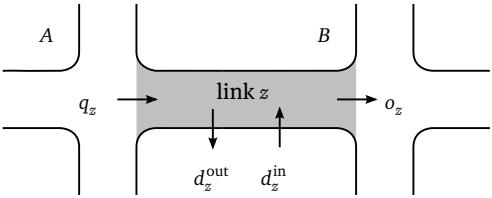


Figure 3.3: Store and Forward Model (adapted from Diakaki et al. 2002)

For each link in the traffic network, as illustrated in Figure 3.3, it is possible to describe the amount of vehicles as a function of time. Given a discrete time k, where $k = 1, 2, 3, \ldots$, and a sample period C (the cycle length), the number of vehicles x_z, inside an arbitrary link z, is dependent on the inflow of vehicles q_z, the outflow o_z, and additional disturbances d_z^{in} and d_z^{out}, that may be depicted as garages and parking lots:

$$x_z(k+1) = x_z(k) + C[q_z(k) - o_z(k) + d_z^{\text{in}}(k) - d_z^{\text{out}}(k)] \tag{3.19}$$

The inflow q_z may be directly collected from detectors when the respective links are the entrance links of the network. For the other links located inside the network, q_z is written as a function of the cycle length C and the outflows of the converging links, according to their green times, saturation flows and turning rates. The outflow o_z follows the same pattern:

$$q_z(k) = \sum_{w \in \mathcal{L}_z} \frac{t^{\uparrow}_{w,z} q^{\text{sat}}_w p_w(k)}{C} \tag{3.20}$$

$$o_z(k) = \frac{q^{\text{sat}}_z p_z(k)}{C} \tag{3.21}$$

where:

- $\mathcal{L}_z$ set containing all the links that converge to link z
- $t^{\uparrow}_{w,z}$ turning rate of link w to z
- q^{sat}_i saturation flow of link i, [veh/s]
- p_i green time for the phase that gives right-of-way to link i, [s]

Substituting Equations 3.20 and 3.21 in Equation 3.19, and assuming that the disturbances d^{in}_z and d^{out}_z are negligible:

$$x_z(k+1) = x_z(k) - q^{\text{sat}}_z p_z(k) + \sum_{w \in \mathcal{L}_z} t^{\uparrow}_{w,z} q^{\text{sat}}_w p_w(k) \tag{3.22}$$

leads to the typical form of the link traffic flow model. It is important to point out that this model is clearly incapable of capturing platooning effects, queue spillbacks and more important, it is only mathematically valid for saturated cases where the flow of vehicles during the green time is kept at the saturation flow level of the given link, i.e. it is integrally used to serve a discharging queue. But this does not exclude its applicability in under-saturated traffic conditions, it just means that the modelling error is greater at these conditions. As will be explained in Section 3.4.2.1 Model Predictive Control, this error was minimized with the introduction of new variables to the model.

Even though x has been used to represent the amount of vehicles inside a link, it is possible to interpret it as the amount of vehicles standing in queue at the time instant just before the related traffic signal turns green, so that the vehicles that do not stand in queue are represented by the inflow of vehicles coming from the converging links. For this reason, x will be referred to as either the queue size or the total amount of vehicles inside the link.

By inspecting Equation 3.22, it is possible to understand that, given its simplicity, the only viable goal for the Green Control Module is to reduce the amount of vehicles in

each link of the network through the choice of the amount of green of the corresponding phases.

3.4.2 Control Problem

The original TUC implementation makes use of the Linear Quadratic Regulator (LQR) to solve the traffic control problem. The LQR comes from the optimal control theory (Kirk 2004) and is essentially a feedback controller that calculates the control actions based on the minimization of a quadratic function and accompanying linear differential equations that describe the system's dynamics. The advantage of using the LQR is that the solution may be calculated offline, regardless of the number of variables, which can be translated, in this case, to the size of the network. For more details see Appendix A.1 LQR.

Nevertheless, the use of a more flexible control technique, called Model Predictive Control (MPC), was proposed by de Oliveira and Camponogara 2007 as a substitute for the LQR in the TUC strategy, and has been confirmed as a better alternative by Aboudolas et al. 2009, 2010. The MPC framework allows the direct implementation of constraints, which are taken into account during the calculation of the control actions. The constraints are basically the maximum and minimum green times allowed in the network, which in the former approach had to be corrected by an extra consecutive step. Other constraints that may be implemented are: rate of change of green times; and maximum number of vehicles allowed in one link, so that a gating effect may prevent spillbacks and gridlocks by automatically limiting the green times of the links converging to the bottlenecks in the network. This gating effect is actually inherent to the traffic model presented, but its influence may be potentialized by limiting the amount of vehicles allowed in one link.

3.4.2.1 Model Predictive Control

MPC is a control technique that consists of an open-loop prediction of the system state by the explicit use of its model. The prediction occurs on a rolling-horizon time frame in which the control actions for each future interval are calculated with the minimization of a cost function. At each run of the algorithm, only the control action of the next interval is implemented and the rest of them gets discarded (Camacho and Bordons 2004). Just to illustrate the idea, imagine a fictitious system, represented by y, being controlled at instant time k as depicted in Figure 3.4. Suppose that the objective is to make y reach a given reference r. The MPC would then find the best set of control actions u that leads the system to r by predicting how the system would respond ($\hat{y}$)

in a horizon of N_p intervals. In the present implementation there is no distinction between the process prediction interval N_p and the control prediction interval N_c, which would usually be the case. Since the control variable used is the amount of green given, it would make no sense to have a control interval N_c smaller than N_p, outside which the green value would be zero.

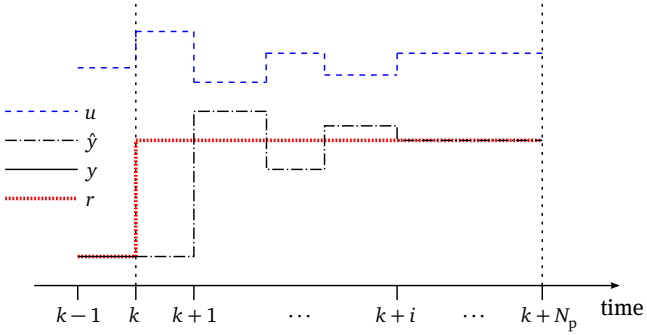


Figure 3.4: Rolling horizon (adapted from CAMACHO and BORDONS 2004)

The quadratic cost function to be minimized, which represents the above given scenario, can, in a simplified way, be expressed as:

$$J = W(\hat{y} - r)^2 + Vu^2 \tag{3.23}$$

where W and V are the gains used to balance the importance of each goal of the cost function: bring $\hat{y}$ as close as possible to r; and minimize the control action u needed. This depicts a single input/single output (SISO) scenario but may be easily extended to a multiple input/multiple output (MIMO) case.

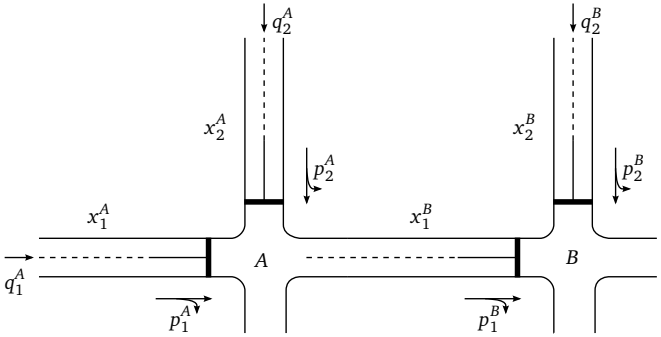


Figure 3.5: Simple network example

Seeking a better understanding of how MPC is applied to the traffic control problem in the present work, an example of a simple two intersection network is presented in Figure 3.5. And the store-and-forward model is used to describe the dynamics of each link of the network:

$$x_1^A(k+1) = x_1^A(k) - q^{\text{sat}}_{x_1^A} p_1^A(k) + C q_1^A \tag{3.24}$$

$$x_2^A(k+1) = x_2^A(k) - q^{\text{sat}}_{x_2^A} p_2^A(k) + C q_2^A \tag{3.25}$$

$$x_1^B(k+1) = x_1^B(k) + t^{\curvearrowright}_{x_1^A, x_1^B} q^{\text{sat}}_{x_1^A} p_1^A(k) + t^{\curvearrowright}_{x_2^A, x_1^B} q^{\text{sat}}_{x_2^A} p_2^A(k) - q^{\text{sat}}_{x_1^B} p_1^B(k) \tag{3.26}$$

$$x_2^B(k+1) = x_2^B(k) - q^{\text{sat}}_{x_2^B} p_2^B(k) + C q_2^B \tag{3.27}$$

which, in turn, can be rewritten in matrix form as the state-space representation of the system:

$$\begin{aligned} X(k+1) &= \boldsymbol{A} X(k) + \boldsymbol{B} u(k) + \boldsymbol{H_q} \\ Y(k) &= \boldsymbol{D} X(k) \end{aligned} \tag{3.28}$$

given:

$$X(k) = \begin{bmatrix} x_1^A(k) & x_2^A(k) & x_1^B(k) & x_2^B(k) \end{bmatrix}^\intercal$$

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{B} = \begin{bmatrix} -q^{\text{sat}}_{x_1^A} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -q^{\text{sat}}_{x_2^A} & 0 & 0 & 0 & 0 & 0 & 0 \\ t^{\curvearrowright}_{x_1^A, x_1^B} q^{\text{sat}}_{x_1^A} & t^{\curvearrowright}_{x_2^A, x_1^B} q^{\text{sat}}_{x_2^A} & -q^{\text{sat}}_{x_1^B} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -q^{\text{sat}}_{x_2^B} & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$u(k) = \begin{bmatrix} p_1^A(k) & p_2^A(k) & p_1^B(k) & p_2^B(k) & s_1^A(k) & s_2^A(k) & s_1^B(k) & s_2^B(k) \end{bmatrix}^\intercal$$

$$\boldsymbol{H_q} = \begin{bmatrix} C q_1^A \\ C q_2^A \\ 0 \\ C q_2^B \end{bmatrix}$$

$$\boldsymbol{D} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where $X(k)$ is the state-variable vector, and $u(k)$ is the control-variable vector, which contains not only the green time p of each phase, but also the green time of each stage s, with:

$$\begin{array}{cc} p_1^A \in s_1^A & p_2^A \in s_2^A \\ p_1^B \in s_1^B & p_2^B \in s_2^B \end{array}$$

Later, in the description of the minimization problem, phases are allowed to assume zero values, and their upper limit is equal to the size of the stage it belongs to. Phases belonging to the same stage are only granted with the necessary amount of green, which keeps the model valid even in low demand conditions (with the LQR formulation this was not possible because the control actions cannot be bounded). Stages must add up to the size of the cycle length, and since they are also a variable of the model, they may grow accompanying a more demanding phase until they reach the minimum of the other stages. After the solution of the minimization problem, the implemented green times are actually the resulting stage lengths.

The cost function minimization must take into account the whole prediction horizon, thus the future values of X and Y must be calculated:

$$\begin{aligned} X(k+1) &= \boldsymbol{A}X(k) + \boldsymbol{B}u(k) + \boldsymbol{H_q} \\ X(k+2) &= \boldsymbol{A}X(k+1) + \boldsymbol{B}u(k+1) + \boldsymbol{H_q} \\ &\vdots \\ X(k+N_\mathrm{p}) &= \boldsymbol{A}X(k+N_\mathrm{p}-1) + \boldsymbol{B}u(k+N_\mathrm{p}-1) + \boldsymbol{H_q} \end{aligned} \tag{3.29}$$

$$\begin{aligned} Y(k) &= \boldsymbol{D}X(k) \\ \hat{Y}(k+1) &= \boldsymbol{D}X(k+1) \\ &\vdots \\ \hat{Y}(k+N_\mathrm{p}) &= \boldsymbol{D}X(k+N_\mathrm{p}) \end{aligned} \tag{3.30}$$

And since the prediction is based on the present time k, all the above Equations are

written based on $X(k)$, for now on represented as $\boldsymbol{X_k}$:

$$X(k+1|k) = \boldsymbol{A}\boldsymbol{X_k} + \boldsymbol{B}u(k) + \boldsymbol{H_q}$$

$$\begin{aligned} X(k+2|k) &= \boldsymbol{A}X(k+1) + \boldsymbol{B}u(k+1) + \boldsymbol{H_q} \\ &= \boldsymbol{A}[\boldsymbol{A}\boldsymbol{X_k} + \boldsymbol{B}u(k) + \boldsymbol{H_q}] + \boldsymbol{B}u(k+1) + \boldsymbol{H_q} \\ &= \boldsymbol{A}^2\boldsymbol{X_k} + \boldsymbol{A}\boldsymbol{B}u(k) + \boldsymbol{A}\boldsymbol{H_q} + \boldsymbol{B}u(k+1) + \boldsymbol{H_q} \end{aligned}$$

$$\begin{aligned} X(k+3|k) &= \boldsymbol{A}X(k+2) + \boldsymbol{B}u(k+2) + \boldsymbol{H_q} \\ &= \boldsymbol{A}[\boldsymbol{A}^2\boldsymbol{X_k} + \boldsymbol{A}\boldsymbol{B}u(k) + \boldsymbol{A}\boldsymbol{H_q} + \boldsymbol{B}u(k+1) + \boldsymbol{H_q}] \\ &\quad + \boldsymbol{B}u(k+2) + \boldsymbol{H_q} \\ &= \boldsymbol{A}^3\boldsymbol{X_k} + \boldsymbol{A}^2\boldsymbol{B}u(k) + \boldsymbol{A}^2\boldsymbol{H_q} + \boldsymbol{A}\boldsymbol{B}u(k+1) + \boldsymbol{A}\boldsymbol{H_q} \\ &\quad + \boldsymbol{B}u(k+2) + \boldsymbol{H_q} \end{aligned}$$

$$\Rightarrow X(k+j|k) = \boldsymbol{A}^j\boldsymbol{X_k} + \sum_{p=0}^{j-1}\boldsymbol{A}^{j-p-1}\boldsymbol{B}u(k+p) + \sum_{p=0}^{j-1}\boldsymbol{A}^{j-p-1}\boldsymbol{H_q} \tag{3.31}$$

$$\Rightarrow \hat{Y}(k+j|k) = \boldsymbol{D}\boldsymbol{A}^j\boldsymbol{X_k} + \sum_{p=0}^{j-1}\boldsymbol{D}\boldsymbol{A}^{j-p-1}\boldsymbol{B}u(k+p) + \sum_{p=0}^{j-1}\boldsymbol{D}\boldsymbol{A}^{j-p-1}\boldsymbol{H_q} \tag{3.32}$$

which can also be put in matrix form:

$$\begin{bmatrix} \hat{Y}(k+1|k) \\ \hat{Y}(k+2|k) \\ \hat{Y}(k+3|k) \\ \vdots \\ \hat{Y}(k+N_\mathrm{p}|k) \end{bmatrix} = \begin{bmatrix} \boldsymbol{D}\boldsymbol{A} \\ \boldsymbol{D}\boldsymbol{A}^2 \\ \boldsymbol{D}\boldsymbol{A}^3 \\ \vdots \\ \boldsymbol{D}\boldsymbol{A}^{N_\mathrm{p}} \end{bmatrix} \boldsymbol{X_k} +$$

$$\begin{bmatrix} \boldsymbol{D}\boldsymbol{B} & \boldsymbol{0}_{n\times m} & \boldsymbol{0}_{n\times m} & \cdots & \boldsymbol{0}_{n\times m} \\ \boldsymbol{D}\boldsymbol{A}\boldsymbol{B} & \boldsymbol{D}\boldsymbol{B} & \boldsymbol{0}_{n\times m} & \cdots & \boldsymbol{0}_{n\times m} \\ \boldsymbol{D}\boldsymbol{A}^2\boldsymbol{B} & \boldsymbol{D}\boldsymbol{A}\boldsymbol{B} & \boldsymbol{D}\boldsymbol{B} & & \boldsymbol{0}_{n\times m} \\ \vdots & \vdots & & \ddots & \\ \boldsymbol{D}\boldsymbol{A}^{N_\mathrm{p}-1}\boldsymbol{B} & \boldsymbol{D}\boldsymbol{A}^{N_\mathrm{p}-2}\boldsymbol{B} & \boldsymbol{D}\boldsymbol{A}^{N_\mathrm{p}-3}\boldsymbol{B} & \cdots & \boldsymbol{D}\boldsymbol{B} \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \\ u(k+2) \\ \vdots \\ u(k+N_\mathrm{p}-1) \end{bmatrix} +$$

$$\begin{bmatrix} \boldsymbol{DH_q} & \mathbf{0}_{n\times 1} & \mathbf{0}_{n\times 1} & \cdots & \mathbf{0}_{n\times 1} \\ \boldsymbol{DAH_q} & \boldsymbol{DH_q} & \mathbf{0}_{n\times 1} & \cdots & \mathbf{0}_{n\times 1} \\ \boldsymbol{DA^2H_q} & \boldsymbol{DAH_q} & \boldsymbol{DH_q} & & \mathbf{0}_{n\times 1} \\ \vdots & \vdots & & \ddots & \\ \boldsymbol{DA^{N_p-1}H_q} & \boldsymbol{DA^{N_p-2}H_q} & \boldsymbol{DA^{N_p-3}H_q} & \cdots & \boldsymbol{DH_q} \end{bmatrix} \tag{3.33}$$

where:

$\mathbf{0}_{i\times j}$ zero matrix with i rows and j columns

n number of state variables (number of links)

m number of control variables (number of phases + number of stages)

and further summarized to:

$$\hat{Y} = \begin{bmatrix} \hat{Y}(k+1|k) \\ \hat{Y}(k+2|k) \\ \hat{Y}(k+3|k) \\ \vdots \\ \hat{Y}(k+N_p|k) \end{bmatrix} \qquad \boldsymbol{M} = \begin{bmatrix} \boldsymbol{DA} \\ \boldsymbol{DA^2} \\ \boldsymbol{DA^3} \\ \vdots \\ \boldsymbol{DA^{N_p}} \end{bmatrix}$$

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{DB} & \mathbf{0}_{n\times m} & \mathbf{0}_{n\times m} & \cdots & \mathbf{0}_{n\times m} \\ \boldsymbol{DAB} & \boldsymbol{DB} & \mathbf{0}_{n\times m} & \cdots & \mathbf{0}_{n\times m} \\ \boldsymbol{DA^2B} & \boldsymbol{DAB} & \boldsymbol{DB} & & \mathbf{0}_{n\times m} \\ \vdots & \vdots & & \ddots & \\ \boldsymbol{DA^{N_p-1}B} & \boldsymbol{DA^{N_p-2}B} & \boldsymbol{DA^{N_p-3}B} & \cdots & \boldsymbol{DB} \end{bmatrix} \qquad u = \begin{bmatrix} u(k) \\ u(k+1) \\ u(k+2) \\ \vdots \\ u(k+N_p-1) \end{bmatrix}$$

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{DH_q} & \mathbf{0}_{n\times 1} & \mathbf{0}_{n\times 1} & \cdots & \mathbf{0}_{n\times 1} \\ \boldsymbol{DAH_q} & \boldsymbol{DH_q} & \mathbf{0}_{n\times 1} & \cdots & \mathbf{0}_{n\times 1} \\ \boldsymbol{DA^2H_q} & \boldsymbol{DAH_q} & \boldsymbol{DH_q} & & \mathbf{0}_{n\times 1} \\ \vdots & \vdots & & \ddots & \\ \boldsymbol{DA^{N_p-1}H_q} & \boldsymbol{DA^{N_p-2}H_q} & \boldsymbol{DA^{N_p-3}H_q} & \cdots & \boldsymbol{DH_q} \end{bmatrix}$$

$$\Rightarrow \hat{Y} = \boldsymbol{MX_k} + \boldsymbol{F}u + \boldsymbol{H} \tag{3.34}$$

Equation 3.34 represents the evolution of the controlled system variables during the prediction interval $[k+1,\ldots,k+N_p]$, and may now be used in Equation 3.23 to complete

the description of the problem to be solved:

$$\boldsymbol{R} = r_x[\overbrace{1111\ldots1}^{nN_\mathrm{p}}]^\intercal \tag{3.35}$$

$$J = \left(\hat{Y} - \boldsymbol{R}\right)^\intercal \boldsymbol{W} \left(\hat{Y} - \boldsymbol{R}\right) + u^\intercal \boldsymbol{V} u \tag{3.36}$$

$$= \left(\boldsymbol{M}\boldsymbol{X}_\boldsymbol{k} + \boldsymbol{F}u + \boldsymbol{H} - \boldsymbol{R}\right)^\intercal \boldsymbol{W} \left(\boldsymbol{M}\boldsymbol{X}_\boldsymbol{k} + \boldsymbol{F}u + \boldsymbol{H} - \boldsymbol{R}\right) + u^\intercal \boldsymbol{V} u \tag{3.37}$$

The dimension of the reference vector $\boldsymbol{R}$ is equal to the number of rows of matrix $\boldsymbol{D}$, nN_p, which maps the state variables being controlled multiplied by the size of the prediction horizon. Its magnitude, r_x, may be set to zero, since the goal of the algorithm is to reduce the amount of vehicles inside the links. Matrix $\boldsymbol{W}$ may be depicted as a block diagonal matrix:

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{W}_\boldsymbol{x} & \boldsymbol{0}_{n\times n} & \boldsymbol{0}_{n\times n} & \cdots & \boldsymbol{0}_{n\times n} \\ \boldsymbol{0}_{n\times n} & \boldsymbol{W}_\boldsymbol{x} & \boldsymbol{0}_{n\times n} & \cdots & \boldsymbol{0}_{n\times n} \\ \boldsymbol{0}_{n\times n} & \boldsymbol{0}_{n\times n} & \boldsymbol{W}_\boldsymbol{x} & & \boldsymbol{0}_{n\times n} \\ \vdots & & & \ddots & \\ \boldsymbol{0}_{n\times n} & \boldsymbol{0}_{n\times n} & \cdots & \boldsymbol{0}_{n\times n} & \boldsymbol{W}_\boldsymbol{x} \end{bmatrix} \tag{3.38}$$

where $\boldsymbol{W}_\boldsymbol{x}$ is also diagonal, and is repeated according to the size of N_p. Returning to the example in Figure 3.5, the matrix $\boldsymbol{W}_\boldsymbol{x}$ would have the following structure:

$$\boldsymbol{W}_\boldsymbol{x} = \begin{bmatrix} \left(x_1^{A,\max}\right)^{-2} & 0 & 0 & 0 \\ 0 & \left(x_2^{A,\max}\right)^{-2} & 0 & 0 \\ 0 & 0 & \left(x_1^{B,\max}\right)^{-2} & 0 \\ 0 & 0 & 0 & \left(x_2^{B,\max}\right)^{-2} \end{bmatrix} \tag{3.39}$$

The gain $\left(x_z^{\max}\right)^{-2}$ normalizes the effect that each queue incurs to the system. Dividing the number of vehicles of a given link by its maximum capacity guarantees that shorter links with the same amount of vehicles of longer links get priority over the latter because they are actually relatively more loaded. Matrix $\boldsymbol{V}$, on the other hand, may be simply described as an identity matrix, multiplied by a gain v_u, which is used to regulate the strength of the control actions. Its dimension is equal to the number of control variables multiplied by N_p.

In Equation 3.37, $\boldsymbol{M}\boldsymbol{X}_\boldsymbol{k}$, $\boldsymbol{H}$ and $\boldsymbol{R}$ are constant matrices and a little work around is done in order to put it in the form expected from a quadratic problem solver:

$$\boldsymbol{L} = \boldsymbol{R} - \boldsymbol{M}\boldsymbol{X}_\boldsymbol{k} - \boldsymbol{H}$$

$$\Rightarrow J = (\boldsymbol{F}u - \boldsymbol{L})^\intercal \boldsymbol{W}(\boldsymbol{F}u - \boldsymbol{L}) + u^\intercal \boldsymbol{V}u$$
$$= u^\intercal \boldsymbol{F}^\intercal \boldsymbol{W}\boldsymbol{F}u - u^\intercal \boldsymbol{F}^\intercal \boldsymbol{W}\boldsymbol{L} - \boldsymbol{L}^\intercal \boldsymbol{W}\boldsymbol{F}u + \boldsymbol{L}^\intercal \boldsymbol{W}\boldsymbol{L} + u^\intercal \boldsymbol{V}u$$

and since $\boldsymbol{W}$ is a diagonal Matrix $\boldsymbol{W}^\intercal = \boldsymbol{W}$, thus $\left(u^\intercal \boldsymbol{F}^\intercal \boldsymbol{W}\boldsymbol{L}\right)^\intercal = \boldsymbol{L}^\intercal \boldsymbol{W}^\intercal \boldsymbol{F}u = \boldsymbol{L}^\intercal \boldsymbol{W}\boldsymbol{F}u$

$$\Rightarrow J = u^\intercal \boldsymbol{F}^\intercal \boldsymbol{W}\boldsymbol{F}u - 2\boldsymbol{L}^\intercal \boldsymbol{W}\boldsymbol{F}u + \boldsymbol{L}^\intercal \boldsymbol{W}\boldsymbol{L} + u^\intercal \boldsymbol{V}u$$

For the minimization procedure the constant values are omitted:

$$J = u^\intercal(\boldsymbol{F}^\intercal \boldsymbol{W}\boldsymbol{F} + \boldsymbol{V})u - 2\boldsymbol{L}^\intercal \boldsymbol{W}\boldsymbol{F}u + \cancel{\boldsymbol{L}^\intercal \boldsymbol{W}\boldsymbol{L}}^{\text{constant}}$$
$$\Rightarrow J = u^\intercal(\boldsymbol{F}^\intercal \boldsymbol{W}\boldsymbol{F} + \boldsymbol{V})u - 2\boldsymbol{L}^\intercal \boldsymbol{W}\boldsymbol{F}u$$

and making $\boldsymbol{Q} = 2(\boldsymbol{F}^\intercal \boldsymbol{W}\boldsymbol{F} + \boldsymbol{V})$ and $\boldsymbol{g} = -2\boldsymbol{L}^\intercal \boldsymbol{W}\boldsymbol{F}$:

$$J = \frac{1}{2} u^\intercal \boldsymbol{Q}u + \boldsymbol{g}u \tag{3.40}$$

which is the final form expected by the solver, where $\boldsymbol{Q}$ and $\boldsymbol{g}$ are the input data and u the solution of the quadratic minimization problem:

$$\begin{aligned}
&\underset{s,p}{\text{minimize}} && J \\
&\text{subject to:} && 0 \leqslant p_z \leqslant \sum_{i \in \mathcal{S}_{p_z}} s_i + \sum_{j \in \mathcal{I}_{p_z}} L_j, \ (\forall z \in \mathcal{P}) \\
& && s_z^{\min} \leqslant s_z \leqslant s_z^{\max}, \ (\forall z \in \mathcal{S}) \\
& && \sum_{j \in \mathcal{S}_{\mathcal{J}_n}} \left(s_j(k) + L_j\right) = C^{\text{trans}}_{\mathcal{J}_n}, \ (\forall n \in \mathcal{J}) \\
& && \sum_{j \in \mathcal{S}_{\mathcal{J}_n}} \left(s_j(k+i-1) + L_j\right) = C^{\text{ref}}, \ (\forall n \in \mathcal{J}; i = 2 \ldots N_{\text{p}})
\end{aligned} \tag{3.41}$$

where:

$\mathcal{S}_{p_z}$	set containing the stages encompassed by phase z
$\mathcal{I}_{p_z}$	set containing the interstages encompassed by phase z
$\mathcal{P}$	set containing all phases of the network
$\mathcal{S}$	set containing all stages of the network
$s_z^{\max}$	maximum available green time for stage z, [s]
$s_z^{\min}$	minimum available green time for stage z, [s]
$C^{\text{trans}}_{\mathcal{J}_n}$	transition cycle of junction n

This optimization problem is solved at each cycle with updated values for $\boldsymbol{X_k}$ and $\boldsymbol{H_q}$. These fresh values are derived from a traffic state estimator that converts the readings from detector loops on each link to number of vehicles.

Even though it hasn't been shown, there might be some cases where it is necessary to fix the length of some stages, e.g. pedestrian stages. These cases are easily overcome by adding an extra set of constraints to the Minimization Problem 3.41. It is just enough to state that the related stages have their upper and lower bounds equal to their fixed lengths:

$$s_z^{\text{fix}} \leqslant s_z \leqslant s_z^{\text{fix}}, \ (\forall z \in \mathcal{S}^{\text{fix}}) \tag{3.42}$$

where:

s_z^{fix} fixed length of stage z, [s]

$\mathcal{S}^{\text{fix}}$ set containing the stages with fixed lengths

MPC has also been recently used in other alternatives for online urban traffic control, as in Kamal et al. 2012; Le et al. in press 2013; Lin et al. 2011, where they have proposed different traffic models to be mapped into the MPC framework. Their goal was more focused in applying a better traffic model, capable of better representing the formation of queues and/or how the traffic dynamics take place. Unfortunately, they were not able to offer an alternative that allows the expected integrated control of all three traffic variables: cycle; offsets; and splits. Lin et al. introduced a macroscopic traffic model, called S Model, which accurately monitors the exchange of traffic volumes between each intersection of the network, and minimizes the total time spent (TTS) in its objective function. Even though it allows the use of different cycle lengths for each intersection, its structure makes it really difficult to account for changing offsets because the update period must be a multiple of the cycle lengths, i.e. the least common multiple of the current cycles lengths across the network, which turns the control strategy unusable during a change of offsets in the network. Kamal et al., in the other hand, presented a traffic model that allows the free change of cycle lengths during the operation of the control strategy. Each intersection cycle is not synchronized with the neighbouring ones by the use of a common cycle length, letting the model's dynamic decide how they progress. The objective function to be minimized maps the amount of vehicles in the network. The presented results show good performance against the reference cases, but it still lacks a better assessment by presenting more realistic scenarios and how it compares, for example, to a network with optimized and synchronized fixed time signal plans. Last but not least, Le et al. proposed a traffic model that better represents the spacial dimension of a traffic network. According to each link's length a compatible number of storing compartments is used to represent the moving of traffic through time. This feature is welcome in cases where links are long and it takes more than a cycle period to cross them. Otherwise, it behaves much like the store-and-forward model used in the current work.

4 | Proposed Modifications to TUC

One of TUC's characteristics is that each control module in its structure operates independently from one another. Particularly, the transition cycles chosen by the Offset Control Module are implemented regardless of their influence in the operation of the network. Analysing the works of Lee and Williams 2009; Pohlmann and Friedrich 2010; Shelby et al. 2006, a potential point for improvement has been identified in TUC's approach. In these works, different techniques for changing the offsets are analysed and compared. They show that the impact of changing the offsets may be reduced by choosing an appropriate transition technique.

Recalling TUC's method for reaching desired offsets in Section 3.3 Offset Control Module, Chapter 3 TUC, the transition cycles may freely range from minimum to maximum cycle lengths regardless of how saturated the network may be. Besides that, because only one transition cycle is allowed, in cases where the transition cycle must be greater than the maximum cycle allowed, the desired offset cannot be fully reached.

Another potential improvement sought was to enable TUC to handle secondary offsets, which may be of great value in meshed networks. In such networks there may be multiple traffic streams crossing each other where their demands vary throughout the day. Opposed to TUC's original objective, it may be of interesting value to allow the onset of desired offsets in the crossing streams which are controlled by the secondary stages in the signal plans.

Summarizing, there were three main reasons that motivated the current investigation:

- The original implementation of offsets in TUC must occur in a single cycle, which may lead to unwanted disruptions in the traffic flow;
- The chosen transition cycles regard only the affected pair of intersections, i.e. their influence on their neighbouring intersections is completely overseen.

- Given TUC's original concept, it is not possible to apply offsets between secondary stages, which would be an interesting feature for mesh networks;

The solution proposed in this work takes advantage of the already available traffic model used by TUC's Green Control Module to calculate the transition cycles. The idea is that by informing the Green Control Module the desired offsets, it will determine the transition cycles that will have less impact in the operation of the network. This also means that the transition cycles may be distributed among the consecutive updates of the control actions, diminishing the disruption caused by the changes.

4.1 Proposed Alternative

4.1.1 Green Control Module Extension

Instead of letting the Offset Control Module blindly calculate the lengths of the transition cycles, the task was transferred to the Green Control Module. The path found to try to circumvent the mentioned imperfections relies on the simple addition of extra equations to the existing set presented in Section 3.4 Green Control Module, i.e. the system of equations used in TUC's Green Control Module that describe the network dynamics. Even though the underlying Store-and-Forward traffic flow model is incapable of modeling the influence of offsets on the behaviour of traffic, the necessary transition cycles do affect the rate at which vehicles are transferred through the network by modifying the length of cycles.

Basically, the idea is to furnish the desired offsets to the Green Control Module and allow it to decide how these offsets will be reached. It will decide which cycles will be extended and which will be shortened taking into consideration how the necessary change in length of each stage may affect the development of the queues as described by its traffic flow model.

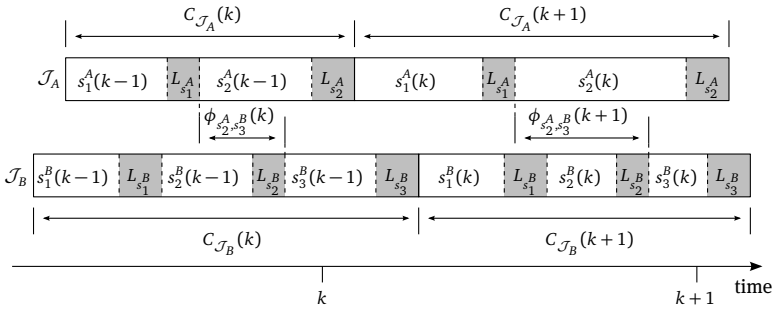


Figure 4.1: Relative Offset

The necessary equations, that allow the proposed approach, may be better understood by looking at Figure 4.1. It depicts the progression in time of two signal plans, one from intersection A and the other from B. It shows the relationship between stages, cycles and offsets at time k, and also what might happen at the next update interval, $k+1$, after new control actions take effect. As one of the goals of the current work was to allow the maintenance of offsets between secondary stages, it is shown what it would be like to describe the relative offset between the second stage of a given intersection A and the third stage of intersection B:

$$\begin{aligned}\phi_{s_2^A,s_3^B}(k+1) = \phi_{s_2^A,s_3^B}(k) + s_3^B(k-1) + L_{s_3^B} + s_1^B(k) + L_{s_1^B} + s_2^B(k) + L_{s_2^B} \\ - \left(s_2^A(k-1) + L_{s_2^A} + s_1^A(k) + L_{s_1^A}\right)\end{aligned} \tag{4.1}$$

Note that this also implies that the resulting cycle lengths will be affected by the stage lengths calculated at time instant k:

$$C_{\mathcal{J}_A}(k+1) = s_1^A(k) + L_{s_1^A} + s_2^A(k) + L_{s_2^A} \tag{4.2}$$

$$C_{\mathcal{J}_B}(k+1) = s_1^B(k) + L_{s_1^B} + s_2^B(k) + L_{s_2^B} + s_3^B(k) + L_{s_3^B} \tag{4.3}$$

which means that not only the offsets must be controlled but also the cycle lengths. Treating offsets and cycles as variables gives an extra degree of freedom to the system. This allows it to choose a better way of reaching these values concerning the impact they may cause in the development of the queues. It also means that these values may be reached in more than one cycle period, proportioning a smoother change in offsets.

With the additional equation set, it is now necessary to bind it together with the existing one. For the sake of clarity, the same example depicted in Figure 3.5, Section

3.4.2.1 Model Predictive Control, is used and the updated equations now read:

$$x_1^A(k+1) = x_1^A(k) - q^{\text{sat}}_{x_1^A} p_1^A(k) + C_{\mathcal{J}_A}(k) q_1^A \tag{4.4}$$

$$x_2^A(k+1) = x_2^A(k) - q^{\text{sat}}_{x_2^A} p_2^A(k) + C_{\mathcal{J}_A}(k) q_2^A \tag{4.5}$$

$$x_1^B(k+1) = x_1^B(k) + t^{\Pi}_{x_1^A, x_1^B} q^{\text{sat}}_{x_1^A} p_1^A(k) + t^{\Pi}_{x_2^A, x_1^B} q^{\text{sat}}_{x_2^A} p_2^A(k) - q^{\text{sat}}_{x_1^B} p_1^B(k) \tag{4.6}$$

$$x_2^B(k+1) = x_2^B(k) - q^{\text{sat}}_{x_2^B} p_2^B(k) + C_{\mathcal{J}_B}(k) q_2^B \tag{4.7}$$

$$\phi_{s_1^A, s_1^B}(k+1) = \phi_{s_1^A, s_1^B}(k) + s_1^B(k) + L_{s_1^B} + s_2^B(k) + L_{s_2^B} - s_1^A(k) - L_{s_1^A} - s_2^A(k) - L_{s_2^A} \tag{4.8}$$

$$\phi_{s_2^A, s_1^B}(k+1) = \phi_{s_2^A, s_1^B}(k) + s_1^B(k) + s_2^B(k) + \sum_{i \in \mathcal{S}_{\mathcal{J}_B}} L_i - s_1^A(k) - s_2^A(k-1) - \sum_{i \in \mathcal{S}_{\mathcal{J}_A}} L_i \tag{4.9}$$

$$C_{\mathcal{J}_A}(k+1) = \sum_{i \in \mathcal{S}_{\mathcal{J}_A}} L_i + \sum_{j \in \mathcal{S}_{\mathcal{J}_A}} s_j(k) \tag{4.10}$$

$$C_{\mathcal{J}_B}(k+1) = \sum_{i \in \mathcal{S}_{\mathcal{J}_B}} L_i + \sum_{j \in \mathcal{S}_{\mathcal{J}_B}} s_j(k) \tag{4.11}$$

This new set of equations may again be written in matrix form as the state-space representation of the system:

$$\begin{aligned} \chi(k+1) &= \boldsymbol{A_k}\chi(k) + \boldsymbol{B}u(k) + \boldsymbol{H_L} \\ Y(k) &= \boldsymbol{D}\chi(k) \end{aligned} \tag{4.12}$$

which is exactly the same structure appearing in Equation 3.28, but with:

$$\chi(k) = \begin{bmatrix} x_1^A(k) & x_2^A(k) & x_1^B(k) & x_2^B(k) & \phi_{s_1^A, s_1^B}(k) & \phi_{s_2^A, s_1^B}(k) & C_{\mathcal{J}_A}(k) & C_{\mathcal{J}_B}(k) & s_2^A(k-1) \end{bmatrix}^{\intercal}$$

$$\boldsymbol{A_k} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & q_1^A & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & q_2^A & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & q_2^B & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{B} = \begin{bmatrix} -q^{\text{sat}}_{x_1^A} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -q^{\text{sat}}_{x_2^A} & 0 & 0 & 0 & 0 & 0 & 0 \\ t^{\curvearrowright}_{x_1^A, x_1^B} q^{\text{sat}}_{x_1^A} & t^{\curvearrowright}_{x_2^A, x_1^B} q^{\text{sat}}_{x_2^A} & -q^{\text{sat}}_{x_1^B} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -q^{\text{sat}}_{x_2^B} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 1 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$u(k) = \begin{bmatrix} p_1^A(k) & p_2^A(k) & p_1^B(k) & p_2^B(k) & s_1^A(k) & s_2^A(k) & s_1^B(k) & s_2^B(k) \end{bmatrix}^{\intercal}$$

$$\boldsymbol{H_L} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \sum_{i \in \mathcal{S}_{\mathcal{J}_B}} L_i - \sum_{i \in \mathcal{S}_{\mathcal{J}_A}} L_i \\ \sum_{i \in \mathcal{S}_{\mathcal{J}_B}} L_i - \sum_{i \in \mathcal{S}_{\mathcal{J}_A}} L_i \\ \sum_{i \in \mathcal{S}_{\mathcal{J}_A}} L_i \\ \sum_{i \in \mathcal{S}_{\mathcal{J}_B}} L_i \\ 0 \end{bmatrix}$$

$$\boldsymbol{D} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Apart from the new offset and cycle variables, it is also necessary to incorporate s_2^A into the state variables in order to keep track of the secondary offset $\phi_{s_2^A, s_1^B}$ during the prediction window. As a general rule, all stages that occur after the stage being used as reference for a given secondary offset, including the stage itself, must be added to the state variable vector, χ. The only exception is when the reference stage is the first one of the cycle.

4.1.1.1 Control Problem

Now that the offsets and cycles have been incorporated into the model, the traffic control problem must be updated accordingly. Just like it was done in Section 3.4.2.1 Model Predictive Control with the queue values, references are passed to the new modeled variables and the goal of the traffic control problem is to reduce the difference between them. The references for the cycles and offsets come from the Cycle and Offset Control Module, as expected. Recalling Equation 3.36:

$$J = \left(\hat{Y} - \boldsymbol{R}\right)^{\intercal} \boldsymbol{W} \left(\hat{Y} - \boldsymbol{R}\right) + u^{\intercal} \boldsymbol{V} u$$

vector $\boldsymbol{R}$ must now incorporate the new references, and, not less important, matrix $\boldsymbol{W}$ must distribute the gains that will guide the decision on how the references will be followed. Vector $\boldsymbol{R}$ is now composed of the concatenation of the references repeated according to the prediction window size N_{p}:

$$\boldsymbol{R}_i = \begin{bmatrix} \boldsymbol{R}_{\boldsymbol{x}} & \boldsymbol{R}_{\boldsymbol{\phi}} & \boldsymbol{R}_{\boldsymbol{C}} \end{bmatrix}^{\intercal}, \ (i = 1 \dots N_{\mathrm{p}}) \tag{4.13}$$

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{R}_1 & \boldsymbol{R}_2 & \boldsymbol{R}_3 & \cdots & \boldsymbol{R}_{N_{\mathrm{p}}} \end{bmatrix}^{\intercal} \tag{4.14}$$

where $\boldsymbol{R}_{\boldsymbol{x}}$ are the queue references, $\boldsymbol{R}_{\boldsymbol{\phi}}$ are the offset references and $\boldsymbol{R}_{\boldsymbol{C}}$ are cycle references. For the specific example in Figure 3.5 they would be:

$$\boldsymbol{R}_{\boldsymbol{x}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \boldsymbol{R}_{\boldsymbol{\phi}} = \begin{bmatrix} \phi^{\mathrm{ref}}_{s_1^A, s_1^B} \\ \phi^{\mathrm{ref}}_{s_2^A, s_1^B} \end{bmatrix} \qquad \boldsymbol{R}_{\boldsymbol{C}} = \begin{bmatrix} C^{\mathrm{ref}} \\ C^{\mathrm{ref}} \end{bmatrix}$$

Similarly, matrix $\boldsymbol{W}$ is also expanded with the gains associated to each set of state variables:

$$\boldsymbol{W}_i = \begin{bmatrix} \boldsymbol{W}_{\boldsymbol{x}} & \boldsymbol{0}_{n_x \times n_\phi} & \boldsymbol{0}_{n_x \times n_C} \\ \boldsymbol{0}_{n_\phi \times n_x} & \boldsymbol{W}_{\boldsymbol{\phi}} & \boldsymbol{0}_{n_\phi \times n_C} \\ \boldsymbol{0}_{n_C \times n_x} & \boldsymbol{0}_{n_C \times n_\phi} & \boldsymbol{W}_{\boldsymbol{C}} \end{bmatrix}, \ (i = 1 \dots N_{\mathrm{p}}) \tag{4.15}$$

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{W}_1 & \mathbf{0}_{n\times n} & \mathbf{0}_{n\times n} & \cdots & \mathbf{0}_{n\times n} \\ \mathbf{0}_{n\times n} & \boldsymbol{W}_2 & \mathbf{0}_{n\times n} & \cdots & \mathbf{0}_{n\times n} \\ \mathbf{0}_{n\times n} & \mathbf{0}_{n\times n} & \boldsymbol{W}_3 & & \mathbf{0}_{n\times n} \\ \vdots & & & \ddots & \\ \mathbf{0}_{n\times n} & \mathbf{0}_{n\times n} & \cdots & \mathbf{0}_{n\times n} & \boldsymbol{W}_{N_\mathrm{p}} \end{bmatrix} \tag{4.16}$$

where $\boldsymbol{W_x}$, $\boldsymbol{W_\phi}$ and $\boldsymbol{W_C}$ for the same example would be:

$$\boldsymbol{W_x} = w_x^\star \begin{bmatrix} \frac{w_{x_1^A}}{\left(x_1^{A,\max}\right)^2} & 0 & 0 & 0 \\ 0 & \frac{w_{x_2^A}}{\left(x_2^{A,\max}\right)^2} & 0 & 0 \\ 0 & 0 & \frac{w_{x_1^B}}{\left(x_1^{B,\max}\right)^2} & 0 \\ 0 & 0 & 0 & \frac{w_{x_2^B}}{\left(x_2^{B,\max}\right)^2} \end{bmatrix}$$

$$\boldsymbol{W_\phi} = w_\phi^\star \begin{bmatrix} w_{\phi_{s_1^A,s_1^B}} & 0 \\ 0 & w_{\phi_{s_2^A,s_1^B}} \end{bmatrix} \qquad \boldsymbol{W_C} = w_C^\star \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The new gains incorporated in $\boldsymbol{W}$ are used to counterbalance the importance of each different set of goals being minimized in the cost function. This is done by adjusting the ratio $w_x^\star : w_\phi^\star : w_C^\star$, and may be fine tuned for each application. Now, in general terms, the cost function may be read as:

$$J = \|x\|_{\boldsymbol{W_x}} + \|\phi - \phi^{\mathrm{ref}}\|_{\boldsymbol{W_\phi}} + \|C - C^{\mathrm{ref}}\|_{\boldsymbol{W_C}} + \|u\|_{\boldsymbol{V}} \tag{4.17}$$

With the cost function restructured, the minimization problem to be solved by the Green Control Module is also updated:

$$\begin{aligned} \underset{s,p}{\text{minimize}} \quad & J \\ \text{subject to:} \quad & 0 \leqslant p_z \leqslant \sum_{i\in\mathcal{S}_{p_z}} s_i + \sum_{j\in\mathcal{I}_{p_z}} L_j, \ (\forall z \in \mathcal{P}) \\ & s_z^{\min} \leqslant s_z \leqslant s_z^{\max}, \ (\forall z \in \mathcal{S}) \\ & C^{\min} \leqslant \sum_{i\in\mathcal{S}_{\mathcal{J}_n}} \left(s_i + L_i\right) \leqslant C^{\max}, \ (\forall n \in \mathcal{J}) \end{aligned} \tag{4.18}$$

4.1.2 Offset Module

The present work, attaining to TUC's philosophy, seeks the achievement of efficient traffic control by following simple rules. TUC's approach uses a very simple traffic model. This model cannot provide estimates of traffic delay, travel time or number of stops as most of the other existing traffic control solutions can. Therefore, the control actions are solely based on what the employed traffic model can describe, which is the distribution of vehicles across the network. The central idea of TUC is that a balanced accumulation of traffic in all the controlled links will also guarantee that, in general terms, delays as well as travel times will be reduced.

In the same path, the offsets are determined according to the size of the downstream queues. This ensures that just enough time is given for the queues to dissipate before the vehicles, which are coming from the upstream intersections, reach them. This strategy is quite straightforward and eliminates the need of using a more complex solution that would determine offsets by optimizing delays, travel times or number of stops.

For the sake of clarification, the available traffic control solution proposed in Pohlmann 2010, which determines the offsets by optimizing traffic delays, was analysed. The resulting behaviour of traffic being controlled by Pohlmann's ATCS showed a different pattern in different situations. Contradicting TUC's, otherwise, plausible strategy, Pohlmann's offsets were not always ensuring progression for the most demanding traffic stream. In order to better understand the dynamics involved, a simple network with two intersections was built in a microscopic traffic simulation software. This network is quite similar to the example depicted in Figure 4.2, but without the third intersection $\mathcal{J}_C$. Within a fixed traffic demand scenario (with a reasonable amount of traffic being routed from phase p_2^A to p_1^B), fixed time signal plans were generated using Webster's formula for calculating optimal cycle lengths and green time percentages. After that, all possible offsets between the two intersections were tested. Then, the setting with the best results in terms of traffic delays was visually revised. Through this setup it was also possible to choose different schemes for traffic demands and routes in order to observe how the best offset setting found was affecting the dynamics of traffic. With these experiments, it was visually possible to infer that the best offset value (the one that resulted in less overall traffic delays) for a given traffic condition was effectively not prioritizing the progression of the most demanding traffic stream. Measuring the average queues being served by both p_3^A and p_1^B, and calculating what offsets would have been stipulated by TUC's approach for each of the competing streams, it has been realized that the wining offset value would always lie in between these offset values. Furthermore, by changing which traffic stream had the most traffic, the wining offset value also seemed to shift towards it.

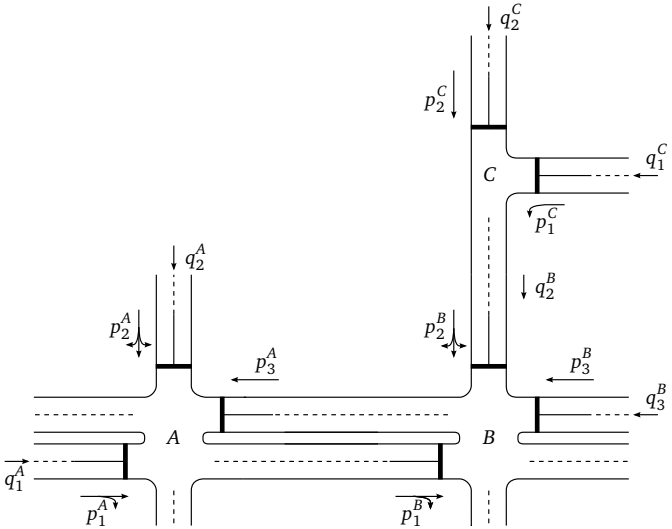


Figure 4.2: Offset example

Without any further proof, and trying to keep the design simple, an unpretentious and unconventional new way of determining the offsets has also been investigated. With the support given by the proposed extensions to the system's model, it was now possible to focus on further alternatives that may now be enforced by the Offset Module. Reviewing TUC's approach to offsets in two way arterials, for each intersection pair, the direction with the highest demand would have been granted with its "desirable" offset, while the other direction would not be considered. Even though this is not common practice, instead of implementing the desirable offset of one of the directions, a mean value of the two conflicting offsets is chosen. Making an analogy to the center of gravity, the offset to be implemented would be the center of mass between both offsets, by substituting the mass with the demand of each traffic stream.

With the extensions proposed earlier in Section 4.1.1 Green Control Module Extension, it is also possible to think of cases where even secondary offsets would influence the final offset. In these cases, as illustrated in Figure 4.2, the expected offset between intersections A and B, $\phi^{\text{ref}}_{A,B}$, would be the "center of demand" of the possible existing offsets:

$$\mathcal{O}_{A,B} = \left\{\left(p_1^A, p_1^B\right), \left(p_2^A, p_1^B\right), \left(p_3^B, p_3^A\right), \left(p_2^B, p_3^A\right)\right\} \tag{4.19}$$

$$q^{\text{tot}}_{\mathcal{O}_{A,B}} = t^{\nearrow}_{p_1^A, p_1^B} q_1^A + t^{\nearrow}_{p_2^A, p_1^B} q_2^A + t^{\nearrow}_{p_3^B, p_3^A} q_3^B + t^{\nearrow}_{p_2^B, p_3^A} q_2^B \tag{4.20}$$

$$\phi_{A,B}^{\text{ref}} = \frac{1}{q_{\mathcal{O}_{A,B}}^{\text{tot}}} \sum_{i \in \mathcal{O}_{A,B}} \phi_i^{\text{ref}} q_i^{\text{r}} \tag{4.21}$$

where:

- $\mathcal{O}_{A,B}$ set containing the available offsets between intersections A and B
- $q_{\mathcal{O}_{A,B}}^{\text{tot}}$ total traffic demand related to the offsets in $\mathcal{O}_{A,B}$, [veh/s]
- $t_{i,j}^{\text{r}}$ turning rate of the traffic flow from phase i to phase j, [-]
- q_i^{r} traffic demand, with the turning rate incorporated, related to offset ϕ_i^{ref}, [veh/s]

Actually, this exact weighting scheme cannot be directly implemented because the secondary offsets have different points of reference, but a similar approach is implemented. Recalling the gain matrix $\boldsymbol{W}$, the reference vector $\boldsymbol{R}$, and their corresponding building blocks $\boldsymbol{W}_\phi$ and $\boldsymbol{R}_\phi$, it is possible to distribute the gains according to the relative demands of each offset. In the example just showed, matrix $\boldsymbol{W}_\phi$ and $\boldsymbol{R}_\phi$ would be:

$$\boldsymbol{W}_\phi = w_\phi^\star \begin{bmatrix} \frac{t^{\text{r}}_{{}^{A}_{p_1},{}^{B}_{p_1}} q_1^A}{q_{\mathcal{O}_{A,B}}^{\text{tot}}} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{t^{\text{r}}_{{}^{A}_{p_2},{}^{B}_{p_1}} q_2^A}{q_{\mathcal{O}_{A,B}}^{\text{tot}}} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{t^{\text{r}}_{{}^{B}_{p_3},{}^{A}_{p_3}} q_3^B}{q_{\mathcal{O}_{A,B}}^{\text{tot}}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{t^{\text{r}}_{{}^{B}_{p_2},{}^{A}_{p_3}} q_2^B}{q_{\mathcal{O}_{A,B}}^{\text{tot}}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{t^{\text{r}}_{{}^{C}_{p_1},{}^{B}_{p_2}} q_1^C}{q_{\mathcal{O}_{B,C}}^{\text{tot}}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{t^{\text{r}}_{{}^{C}_{p_2},{}^{B}_{p_2}} q_2^C}{q_{\mathcal{O}_{B,C}}^{\text{tot}}} \end{bmatrix} \tag{4.22}$$

$$\boldsymbol{R}_{\boldsymbol{\phi}} = \begin{bmatrix} \phi^{\text{ref}}_{{}^{A}_{p_1},{}^{B}_{p_1}} & \phi^{\text{ref}}_{{}^{A}_{p_2},{}^{B}_{p_1}} & \phi^{\text{ref}}_{{}^{B}_{p_3},{}^{A}_{p_3}} & \phi^{\text{ref}}_{{}^{B}_{p_2},{}^{A}_{p_3}} & \phi^{\text{ref}}_{{}^{C}_{p_1},{}^{B}_{p_2}} & \phi^{\text{ref}}_{{}^{C}_{p_2},{}^{B}_{p_2}} \end{bmatrix}^{\intercal} \tag{4.23}$$

which has the same effect that Equation 4.21 would produce when accordingly applied to the other set of offsets $\mathcal{O}_{B,C} = \left\{\left(p_1^C, p_2^B\right), \left(p_2^C, p_2^B\right)\right\}$. The Offset Module is now in charge of not only calculating the desirable offsets but also to determine the gains that will be used by the Green Control Module regarding the offsets.

It is also important to point out that the queue values used in the Offset Module are not exactly the queue lengths present during the switch from red to green of the downstream intersection, but the downstream queue length values at the instant that the upstream traffic signal releases its traffic stream. Take for example the two competing offsets $\phi_{p_1^A,p_1^B}$ and $\phi_{p_2^A,p_1^B}$ from previous example depicted in Figure 4.2. The queue length values in p_1^B are different for each case and must be accounted for during the calculation of the offset. The estimation of these different values is made possible with Queue Predictor presented in the next Chapter.

4.1.2.1 Mesh Networks

Last but not least, it is also important to discuss how mesh networks were treated in the view of offset setting. In mesh networks, like the one depicted in Figure 4.3, the formation of loops between the links makes it impossible to guarantee that the each pair of intersections will be granted with an "ideal" offset. A loop is any possible closed geometrical form that may be constructed with the use of consecutive links in the network. For contemplating all offsets in a loop, it is necessary that the sum of the primary offsets must be a value multiple of the common cycle of the network. This condition is dependent on the geometry of the network and the traffic conditions, which can only be met in fortuitous scenarios.

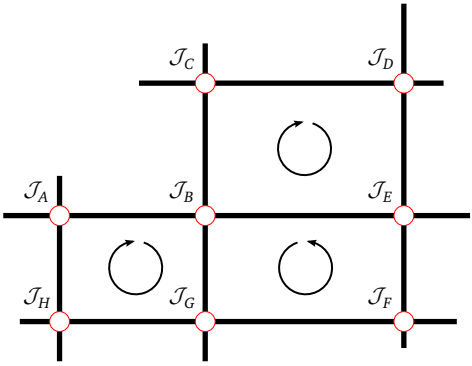


Figure 4.3: Mesh network

Given the available gain structure in matrix W_ϕ it would have been possible to handle conflicting loop offsets by following a similar approach to two way roads, where each traffic stream involved receives a weight proportional to its traffic demand. Two way road segments/links are actually a particular case of a mesh network, with just two

junctions involved. A similar weighting scheme, where the gain is a percentage of the total demand, could be employed for the the road segments in the loop, only if none of the segments is part of another loop. Looking at Figure 4.3, it is possible to identify six different loops: $\{(\overline{AB},\ \overline{BG}, \overline{GH}, \overline{HA}), (\overline{BG},\ \overline{GF}, \overline{FE},\ \overline{EB}), (\overline{CD},\ \overline{DE}, \overline{EB},\ \overline{BC}), (\overline{AB},\ \overline{EB}, \overline{FE}, \overline{GF}, \overline{GH},\ \overline{HA}), (\overline{CD},\ \overline{DE}, \overline{FE}, \overline{GF}, \overline{BG},\ \overline{BC}), (\overline{AB},\ \overline{BC}, \overline{CD}, \overline{DE}, \overline{FE}, \overline{GF}, \overline{GH},\ \overline{HA})\}$, where each loop has at least one segment that belongs to another loop. In these cases, even if the gains (related to each loop the segment belongs to) were concatenated, the weighting scheme would not work as intended. Take for example the segment $\overline{FE}$, it belongs to four of the existing loops. Following the weighting structure used for the internal flows, the resultant gain related to $\overline{FE}$ would be:

$$w_{\phi_{F,E}} = \frac{q^{\text{tot}}_{\mathcal{O}_{F,E}}}{q^{\text{tot}}_{\mathcal{O}_{B,G,F,E}}} \frac{q^{\text{tot}}_{\mathcal{O}_{F,E}}}{q^{\text{tot}}_{\mathcal{O}_{C,D,E,F,G,B}}} \frac{q^{\text{tot}}_{\mathcal{O}_{F,E}}}{q^{\text{tot}}_{\mathcal{O}_{A,B,E,F,G,H}}} \frac{q^{\text{tot}}_{\mathcal{O}_{F,E}}}{q^{\text{tot}}_{\mathcal{O}_{A,B,C,D,E,F,G,H}}} \tag{4.24}$$

Now compare to the expected concatenated gains of the segment $\overline{BG}$, that is part of the same inner loop $(\overline{BG},\ \overline{GF}, \overline{FE},\ \overline{EB})$, but that belongs to just three of the loops:

$$w_{\phi_{G,B}} = \frac{q^{\text{tot}}_{\mathcal{O}_{B,G}}}{q^{\text{tot}}_{\mathcal{O}_{B,G,F,E}}} \frac{q^{\text{tot}}_{\mathcal{O}_{B,G}}}{q^{\text{tot}}_{\mathcal{O}_{C,D,E,F,G,B}}} \frac{q^{\text{tot}}_{\mathcal{O}_{B,G}}}{q^{\text{tot}}_{\mathcal{O}_{A,B,G,H}}} \tag{4.25}$$

Even if their demands were the same, $q^{\text{tot}}_{\mathcal{O}_{B,G}} = q^{\text{tot}}_{\mathcal{O}_{F,E}}$, segment $\overline{GB}$ would have priority over $\overline{FE}$ because its final gain $w_{\phi_{G,B}}$ would probably be bigger (remember that each factor of the product is a number in the range $[0.0, 1.0]$). Either way, their relative gains would not be compatible with their relative demands.

Algorithm 4.1: Loop breaker

```
φ : set containing all loops in the network;
foreach φ_i ∈ φ do
    demand ← ∞;
    segmentToEliminate ← NONE;
    foreach segment_j ∈ φ_i do
        if q_segment_j < demand then
            demand ← q_segment_j;
            segmentToEliminate ← segment_j;
    foreach φ_j ∈ segmentToEliminate do
        w_φj ← 0.0;
```

Since the same weighting scheme would not work for all network types, it was left to be used only for calculating the inner gains for each intersection pair. When there are loops, the Algorithm 4.1 Loop breaker is run and the weakest segment of each loop, the one with the lowest demand, is eliminated (by making the associated gains equal

to zero), settling the case.

Associated with the findings from the offsets experiments performed, the possibility of playing with the gains in matrix W_ϕ have led to the proposal of three different offset strategies: Selfish; Democratic; and Partially Democratic, that are enforced by the Offset Control Module.

4.1.2.2 Selfish Strategy

The Selfish Strategy is actually the usual case where only one traffic stream, belonging to a pair of junctions, is granted with the "green-wave". The only difference is that perhaps a secondary offset may be chosen, as long as it has the highest traffic demand. The Offset Module implements this strategy by simply setting the gains that will be used in matrix W_ϕ to zero, for the offsets with lower traffic demands, and to 1.0, for the offsets with the highest demands from each pair of intersections. This strategy plays the role of a reference case to be compared against the others. In the example shown in Figure 4.2, if $t^{\uparrow}_{p_3^B,p_3^A} q_3^B$ and $t^{\uparrow}_{p_2^C,p_2^B} q_2^C$ were the highest demands, the matrix W_ϕ would be:

$$W_\phi = w^\star_\phi \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

4.1.2.3 Democratic Strategy

The Democratic Strategy brings together all the potential made available by the extensions proposed in this investigation. All possible offsets are taken into consideration, and their gains in matrix W_ϕ are adjusted to emulate the "center of demand" effect, and, using the example in Figure 4.2, would assume the exact form presented in Equation 4.22.

4.1.2.4 Partially Democratic Strategy

This Strategy follows the same path of the previous one, but it only grants offset maintenance for the most demanding traffic streams in each pair of intersections. The

offsets included must add up to at least 50% of the associated total traffic demand. Taking the example of Figure 4.2 again, and assuming:

$$t^{\vec{\Gamma}}_{p_1^A,p_1^B} q_1^A < t^{\vec{\Gamma}}_{p_2^B,p_3^A} q_2^B < t^{\vec{\Gamma}}_{p_2^A,p_1^B} q_2^A < t^{\vec{\Gamma}}_{p_3^B,p_3^A} q_3^B$$

$$t^{\vec{\Gamma}}_{p_3^B,p_3^A} q_3^B < \frac{1}{2} q^{\text{tot}}_{\mathcal{O}_{A,B}}$$

$$t^{\vec{\Gamma}}_{p_2^A,p_1^B} q_2^A + t^{\vec{\Gamma}}_{p_3^B,p_3^A} q_3^B > \frac{1}{2} q^{\text{tot}}_{\mathcal{O}_{A,B}}$$

$$t^{\vec{\Gamma}}_{p_1^C,p_2^B} q_1^C > t^{\vec{\Gamma}}_{p_2^C,p_2^B} q_2^C$$

the resulting $\boldsymbol{W}_\phi$ would be:

$$\boldsymbol{W}_\phi = w_\phi^\star \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{t^{\vec{\Gamma}}_{p_2^A,p_1^B} q_2^A}{q^{\text{part}}_{\mathcal{O}_{A,B}}} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{t^{\vec{\Gamma}}_{p_3^B,p_3^A} q_3^B}{q^{\text{part}}_{\mathcal{O}_{A,B}}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

with $q^{\text{part}}_{\mathcal{O}_{A,B}} = t^{\vec{\Gamma}}_{p_2^A,p_1^B} q_2^A + t^{\vec{\Gamma}}_{p_3^B,p_3^A} q_3^B$. And since there are only two competing traffic streams with their associated offsets between intersections $\mathcal{J}_B$ and $\mathcal{J}_C$, the position related to the most demanding one receives gain 1 and the other gain 0.

4.1.3 Unexpected Setback

The new cost function proposed (Equation 4.17) must be syntonized to allow a harmonic balance in the achievement of each of its goals: reduce the number of vehicles inside the network; reach the given offsets; and attain to the desired common cycle length. This is done by tuning the gains in matrix $\boldsymbol{W}$, i.e. adjusting the gains' ratio $w_x^\star : w_\phi^\star : w_C^\star$ and their individual absolute values. The policy chosen was to keep these gains constant and independent from traffic conditions, which allows the control strategy to automatically give more weight to the goal that aims the reduction of the number of vehicles inside the network as its demands increase. Reminding Equation 4.17, the higher the demands, the bigger are the queues and number of vehicles inside, the greater is the factor $\|x\|_{\boldsymbol{W}_x}$ in relation to the others.

After an extensive tuning, the balanced gains found for the ratio $w_x^\star : w_\phi^\star : w_C^\star$ caused an unexpected output whenever there were any secondary offsets active during the minimization of the cost function. In the attempt to conform to the specified secondary offset, the stages before the one being referenced were frequently getting compressed until their minimum values, causing great harm to the operation of the network. This compression effect can be better visualized in the example shown in Figure 4.4. The offset $\phi_{s_2^A,s_3^B}$ may force the stages s_1^A, s_1^B and s_2^B to retract. This effect occurs in undersaturated scenarios, where the relative influence of x is much smaller than ϕ in the output of the cost function, and where the actual green time distributed for the phases by the controller is mostly governed by the size of the cycles other than by the size of the queues. But, it is also observed, independent from traffic conditions, if a conflicting primary offset pushes the beginning of the cycle against the other direction.

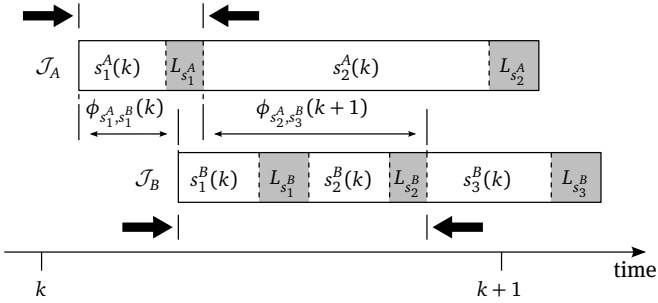


Figure 4.4: Secondary offset compression effect

This problem could have been eliminated by simply reducing the value of $w_\phi^\star$, but this solution required such low values for $w_\phi^\star$ that, at the end, the desired offsets would not be reached in reasonable time. Another discarded solution made use of the available room in the prediction window. Recalling the block structure of matrix $\boldsymbol{W}$, depicted in Equations 4.15 and 4.16, there is one matrix $\boldsymbol{W}_\phi$ for each future interval in the range $[1, N_p]$. Even though they were all supposed to be the same, it is possible to apply a different scheme. The control actions that will be effectively used belong to the first interval of the prediction horizon, $k+1$, but, the remaining intervals may be forced to keep similar conditions to the first one by adding extra constraints to the Control Problem 4.18. In this case, each stage was constrained to be within a given range from the previous interval, so that:

$$s_z(k+i-1) - \Delta t \leqslant s_z(k+i) \leqslant s_z(k+i-1) + \Delta t, \ (\forall z \in \mathcal{S};\ i = 2 \ldots N_p) \tag{4.26}$$

Thus, a given stage, along the consecutive intervals, would not change too much, which would also mean that the offsets would not drift away too much. With this

in mind, setting the gains of W_ϕ to zero in the first interval, $k+1$, would still guarantee that the offsets would reach a value close to their expected ones, and, more importantly, it would alleviate the strain on the green times. Once again, unfortunately, the necessary values for Δt, that would eliminate the stage compression problem, were so big that the offsets, at the end, would not be close to the expected references.

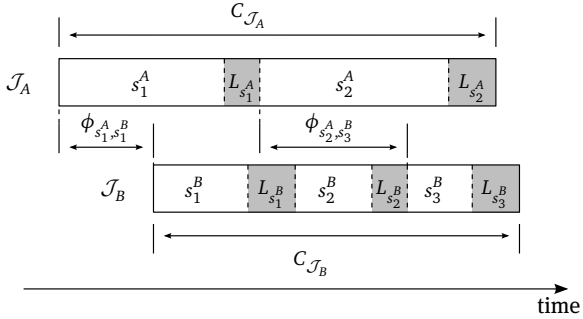


Figure 4.5: Fixed time offset

The workaround found is based on the expected overall maintenance of the average traffic conditions. With fixed time signal plans, it is easy to enforce secondary offsets because the length of every green time remains fixed. In the example depicted in Figure 4.5, which is the fixed time version of the same scenario in Figure 4.1, the offset between stages s_2^A and s_3^B, $\phi_{s_2^A,s_3^B}$, may be enforced by simply setting the primary offset $\phi_{s_1^A,s_1^B}$ equal to:

$$\phi_{s_1^A,s_1^B} = \phi_{s_2^A,s_3^B} + s_1^A + L_{s_1^A} - s_1^B - L_{s_1^B} - s_2^B - L_{s_2^B} \tag{4.27}$$

In the other hand, in an online traffic control system like TUC, the stage lengths change every cycle and a secondary offset may not be kept at the "ideal" point. Even though this is true, it is reasonable to expect that this oscillation occurs around a mean value, and that the secondary offsets in this case would be as much affected as would be the secondary offsets in a fixed time scenario, given the stochastic behaviour of traffic.

Assuming that the traffic conditions are constant during a limited time period, the TUC's calculated green times during this interim will oscillate around a mean value. And this average may be calculated using TUC's past signal plans. In the present work, the past three signal plans are used to infer the next possible green times that will be calculated by the Green Control Module, and a simple average of them is used to determine the primary offset associated to the desired secondary offset. Using the example in Figure 4.1 the desired secondary offset $\phi^{\text{ref}}_{s_2^A,s_3^B}$ would be achieved by calculating the associated approximate primary offset $\phi^{\text{ref}}_{s_1^A,s_1^B|s_2^A,s_3^B}$ with the use of the past

implemented stage lengths:

$$\bar{s}_1^A = \frac{C^{\text{ref}}(k+1)}{3}\left(\frac{s_1^A(k-1)}{C_{\mathcal{J}_A}(k)} + \frac{s_1^A(k-2)}{C_{\mathcal{J}_A}(k-1)} + \frac{s_1^A(k-3)}{C_{\mathcal{J}_A}(k-2)}\right) \tag{4.28}$$

$$\bar{s}_2^A = \frac{C^{\text{ref}}(k+1)}{3}\left(\frac{s_2^A(k-1)}{C_{\mathcal{J}_A}(k)} + \frac{s_2^A(k-2)}{C_{\mathcal{J}_A}(k-1)} + \frac{s_2^A(k-3)}{C_{\mathcal{J}_A}(k-2)}\right) \tag{4.29}$$

$$\bar{s}_1^B = \frac{C^{\text{ref}}(k+1)}{3}\left(\frac{s_1^B(k-1)}{C_{\mathcal{J}_B}(k)} + \frac{s_1^B(k-2)}{C_{\mathcal{J}_B}(k-1)} + \frac{s_1^B(k-3)}{C_{\mathcal{J}_B}(k-2)}\right) \tag{4.30}$$

$$\phi^{\text{ref}}_{s_1^A, s_1^B | s_2^A, s_3^B} \approx \phi^{\text{ref}}_{s_2^A, s_3^B} + \bar{s}_1^A + L_{s_1^A} - \bar{s}_1^B - L_{s_1^B} - \bar{s}_2^B + L_{s_2^B} \tag{4.31}$$

Now that the Offset Module is capable of maintaining secondary offsets with the above presented transformations, the three different subtypes of offset control presented in Section 4.1 Proposed Alternative: Selfish; Democratic; and Partially Democratic may also be implemented. For the last two, the concept of "center of demand" may be directly applied with the use of Equation 4.21, after the secondary offsets are "transformed" in primary offsets. The resulting, final, offset reference found is then passed to the Green Control Module, which will consider traffic conditions and neighbouring offsets for generating the necessary signal plans.

It is important to point out that, in order for this approach to work, it is necessary that both Cycle Control and Offset Control modules have the same update period and that they occur always at the same time. Unlike the Green Control Module, which gets updated at every new cycle, the last two modules define their cycle and offset references for longer periods, in which they remain constant. If the renovation of these references happens at different times, the unsynchronized change of the common cycle length would dislocate the secondary offsets, driving the system further away from the expected operating point.

4.2 Legacy Alternative

The Legacy Alternative has been created in order to attest the usefulness of the former one. The whole concept of the Proposed Alternative in Section 4.1 was aimed at proportioning the ability to consider secondary offsets, and to enforce offsets in general in a way that would disturb the least possible the traffic conditions. Given the limitations of the traffic flow model used (see Section 3.4.1 Traffic Flow Model), it could be possible that it would not be capable of reproducing the disturbances caused by

the changing cycle lengths during the process of implementing new offsets. If this was the case, then the Proposed Alternative could even cause more harm than benefit comparing to TUC's original strategy (one cycle transition plans), and perhaps a much simpler approach would be better.

The first point to tackle was the one cycle transition plans. Instead of redesigning and extending the Green Control Module, the Legacy Alternative only actuates in the Offset Control Module, keeping the rest of TUC's configuration, as presented in Chapter 3, unaltered. Following the discussion in Shelby et al. 2006 and the offset configuration in Pohlmann 2010, the Offset Control Module was modified to limit the transition cycles to ±20% of the current common cycle length, and to keep generating transition plans until the desired offsets were reached, i.e. implementing the *Shortway* method.

In order to deal with secondary offsets, the same approach presented previously in Section 4.1.3 Unexpected Setback is used. The secondary offsets are "transformed" into primary offsets, and depending on the subtype of offset control used (Selfish, Democratic, or Partially Democratic), a final offset is calculated for each pair of intersections. The transition cycles are then calculated successively as described in Section 3.3 Offset Control Module in Chapter 3, and passed as fixed values for the Green Control Module.

5 | Queue Estimation

All ATCSs need some sort of traffic state estimation to be capable of determining the right set of actions that will produce a better operation of the traffic network. For the TUC strategy, it is expected the availability of three types of magnitude: queue size; traffic demand; and turning rate. Even though TUC (Diakaki 1999) has been presented with an estimator for the number of vehicles inside the links, which doubles as queue estimator, a new alternative has been developed and used throughout the current investigation.

In the process of analysing how TUC's Green Control Module functions, it was identified a possible way to improve its performance. TUC's strategy aims the control of the network by trying to respond to the stochastic behaviour of traffic, where the queue lengths and vehicle flows change at each new cycle. In order to justify the frequency at which TUC's control actions are calculated, these changes must be captured. Since these control actions are to be applied on the next cycle, then the simple estimation of the past queue lengths, or even the current amount of vehicles in each link, becomes a suboptimal solution. The ideal scenario would be to predict the queues and, with these future values, calculate the necessary amount of green for each of them.

Looking at a traffic network, it is possible to visualize that the traffic demand of a given link is made up of the traffic flow being let through the upstream links. In the same way, the queues being formed are also a result of the queues that were served by the upstream links. Therefore, if the dynamics of the network, along with the estimated past queue lengths, are taken into consideration, it is possible to better assess how the future queues will develop. And this was the path followed.

The literature presents many alternatives for the estimation of traffic states in urban roads, but none of them really takes advantage of the use of the inherent dynamics of the whole network in order to improve the estimation. There is a range of mathematical tools being used: Markov Chains (Viti and Zuylen 2004; Yu et al. 2003); Neural Networks (Zheng et al. 2006); Kalman Filtering (Gang et al. 2007; Jabari and

Liu 2013; Mück 2002b; Pueboobpaphan and Nakatsuji 2006; Tampère and Immers 2007); and Cell Transmission Model (Chen et al. 2010; Gang et al. 2007; Hu et al. 2010; Jabari and Liu 2012; Tampère and Immers 2007), which confer good results to the solution, but do not directly fit the requirements of the current application. Except the recent work from Jabari and Liu, they either need a much larger interval between each estimation, or just limit the scope to individual road stretches, which in most cases is due to their complexity. Seeking a much simpler approach, Vigos and Papageorgiou 2010 presented a queue estimator much similar to the one in Diakaki 1999, where individual queue lengths are a direct product of the occupancy of single loop detectors situated in the middle of the links.

From the above cited works, Tampère and Immers 2007 is the one that better approximates the envisioned solution. It uses the CTM as a model for the whole network, and, by mirroring the information from detectors into the related cells in the model, it is capable of furnishing the desired traffic states. The disadvantage of this technique is that a much refined traffic information is needed, so that its resolution in time matches the one in the model, e.g. detector occupancy with a one second resolution. Apart from that, the estimation ends up being restrictive in its effectiveness, because only the state estimates of the cells neighbouring the one associated to the real physical detector really profit from it.

Even though the proposed solution resembles the work from Tampère and Immers, it was actually initially inspired by Mück 2002b. Mück divides the estimation problem in two steps: a queue length estimation is made based on the fill-up time of an inductive loop detector located on the link in subject; and then, this "raw" measurement is fed to an Extended Kalman Filter (Anderson and Moore 2012) coupled with a delay and queue length model proposed by Kimber and Hollis 1979. The current solution uses the same strategy. It applies local estimations of queue lengths as measurements for a more complex model attached to an Unscented Kalman Filter (Julier et al. 1995; Julier and Uhlmann 1997). By substituting the model with a CTM encompassing the whole network, it is possible to account for the influence of the dynamics of the network in the formation of the queues. Besides that, a third step is incorporated to the technique, which uses the CTM, updated with the values from the former step, to simulate future developments of the queues and functioning as a queue predictor.

The following sections will describe each component of the proposed queue prediction technique.

5.1 Local Queue Estimation

For the first step of the proposed queue estimator two alternatives were tested. The first one was the original method proposed by MÜCK 2002b, which uses the fill-up time of the detector. And the second one, slightly different, was the local queue estimator proposed by VIGOS and PAPAGEORGIOU 2010, which is based on the detector occupancy.

5.1.1 Mück's Local Queue Estimator

Mück's local queue estimator relates the time that a growing queue takes to reach the inductive loop detector, the fill-up time, and the final length of the queue. The longer it takes, the smaller is the demand and therefore, the smaller is the queue.

Before the estimation can be applied, there is the need to collect two different set of real/actual measurements on the location in subject. The first, of course, is the real queue length, x^{real}, occurring in each cycle. And the second one, is the associated fill-up time of the detector t^{fill}. For better results, the gathering of a reasonable amount of measurements is expected. When possible, the observation period should include different traffic conditions. The value of t^{fill} must be incremented from the instant the signal changes from green to yellow/red until the associated detector reaches full occupancy, or, the signal changes back to green, whichever comes first. In case the detector is not reached by the growing queue, an arbitrary big value must be added to t^{fill}, so that it does not count as "filled". In each successive cycle, a collected measurement pair $\left(x_k^{\text{real}}, t_k^{\text{fill}}\right)$ is time-indexed, k, and stored. Afterwards, the fill-up time is processed by creating the jam parameter τ and applying it to each measurement:

$$\tau_k = \begin{cases} 1 & t_k^{\text{fill}} \leqslant t^{\text{ref}} \\ 0 & t_k^{\text{fill}} > t^{\text{ref}} \end{cases} \tag{5.1}$$

where t^{ref} is the reference fill-up time, which must be tuned for each link. After this preliminary step, both τ and x^{real} are filtered through exponential smoothing and become $\bar{\tau}$ and $\bar{x}$:

$$\bar{\tau}_k = \alpha^{\text{tau}} \tau_k + (1 - \alpha^{\text{tau}}) \bar{\tau}_{k-1} \tag{5.2}$$

$$\bar{x}_k = \alpha^{\text{queue}} x_k^{\text{real}} + (1 - \alpha^{\text{queue}}) \bar{x}_{k-1} \tag{5.3}$$

where α^{tau} and α^{queue} are the smoothing factors, which have a magnitude around 0.1 (MÜCK 2002b).

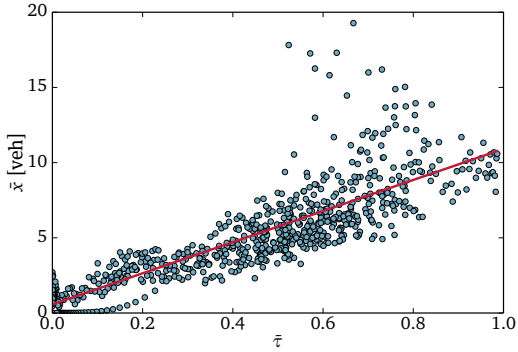


Figure 5.1: Linear fit $\bar{x} \times \bar{\tau}$ (source: DANTAS and FRIEDRICH 2013)

Then, by plotting the smoothed points, e.g. as shown in Figure 5.1, a linear relationship between $\bar{x}$ and $\bar{\tau}$ can be extracted:

$$\bar{x}_k = a\bar{\tau}_k + b \tag{5.4}$$

where a is the slope; and b the axis bias. With a and b calibrated, it is now possible to estimate the queue length, $\hat{x}$, with fresh values of $\bar{\tau}$ being collected online from the detector in the street:

$$\hat{x}_k = a\bar{\tau}_k + b \tag{5.5}$$

5.1.2 Vigos's Local Queue Estimator

After investigating the relationship between a detector's time occupancy and the related link's space occupancy in PAPAGEORGIOU and VIGOS 2008, i.e. the dependency between the number of vehicles inside the link and the resulting percentage of time that part of them occupied the detector during their drive through, VIGOS et al. 2008 proposed a Kalman Filter-based queue estimator. It initially required three measurement points, i.e. three detectors per link being monitored. But, realising that this number of detectors per link would be costly impracticable, they also analysed the results of its proposed queue estimator when fed with just one detector. Given the reasonable results with just one detector, they simplified the queue estimator by substituting the Kalman Filter with an exponential filter in VIGOS and PAPAGEORGIOU 2010, which will be presented next.

The space occupancy, o^{spc}, of a given link is by definition:

$$o^{\text{spc}} = \frac{x\bar{l}^{\text{veh}}}{l^{\text{link}}n^{\text{lanes}}} \tag{5.6}$$

where:

- x number of vehicles inside the link, [veh]
- $\bar{l}^{\text{veh}}$ expected average physical vehicle length, [m]
- l^{link} link's length, [m]
- n^{lanes} number of lanes

and it may be related to the measured time occupancy, o^{time}, collected during the time interval
$[(k-1)\Delta t\ ,\ k\Delta t)$, as:

$$o^{\text{time}}(k-1) = o^{\text{spc}}(k-1) + \zeta_0(k-1) \tag{5.7}$$

with ζ_0 incorporating different sources of error: measurement errors; impact of the measurement frequency; and error due to the different vehicle lengths.

If the relation of the time occupancy and the measured vehicle count, x^{msr}, is described as:

$$o^{\text{time}} = \frac{\bar{l}^{\text{veh}}}{l^{\text{link}}n^{\text{lanes}}}x^{\text{msr}} \tag{5.8}$$

then, substituting Equations 5.6 and 5.8 in 5.7:

$$\zeta = \frac{\zeta_0 l^{\text{link}}n^{\text{lanes}}}{\bar{l}^{\text{veh}}} \tag{5.9}$$

$$x^{\text{msr}}(k-1) = x(k-1) + \zeta(k-1) \tag{5.10}$$

it is possible to state that the measured number of vehicles is the actual number of vehicles under the presence of noise. Note that $(l^{\text{link}}n^{\text{lanes}})/\bar{l}^{\text{veh}}$ is actually the maximum number of vehicles, x^{max}, that would fit inside the link in a bumper to bumper manner. Since this does not actually happen in real life, a correction term, ϱ, is introduced in order to retain the real expected jam density of the link in subject:

$$x^{\text{max}} = \frac{l^{\text{link}}n^{\text{lanes}}}{\left(\bar{l}^{\text{veh}} + \varrho\right)} \tag{5.11}$$

and is usually set to 1 meter. Another correction term, proposed in Papageorgiou and Vigos 2008, accounts for the probable detector length (considering inductance loop detectors) that directly interferes with the measured period of time during which the detector is occupied by a passing vehicle. Therefore, the final value used for the

time occupancy is:

$$\hat{o}^{\text{time}} = o^{\text{time}} \frac{\bar{l}^{\text{veh}}}{\bar{l}^{\text{veh}} + \varepsilon} \tag{5.12}$$

with ε being the correction term, and made equal to the real detector length. After compensating for these two sources of error, Equation 5.8 is rewritten and the measured number of vehicles is now expressed as:

$$x^{\text{msr}} = \hat{o}^{\text{time}} x^{\max} \tag{5.13}$$

The resulting queue estimator is an exponential filter that weights the past estimated value and the last measurement:

$$\hat{x}_k = \alpha^{\text{SM}} x_{k-1}^{\text{msr}} + \left(1 - \alpha^{\text{SM}}\right) \hat{x}_{k-1} \tag{5.14}$$

with α^{SM} being the smoothing factor in the range $[0, 1]$. In Vigos and Papageorgiou 2010, a comprehensive analysis on how to choose this smoothing parameter is presented. Basically, they show how the estimation is affected by: the choice of traffic signal control being used, fixed or adaptive, i.e. how it responds to changing cycle lengths; the update period of the estimator, i.e. equal to or smaller than the cycle length; the length of the link; and also by fixed and varying traffic demands.

5.2 Cell Transmission Model

Proposed by Daganzo 1994, 1995, the Cell Transmission Model (CTM) has been derived from the established Lighthill-Whitham-Richards (LWR) model (Lighthill and Whitham 1955; Richards 1956), which is a macroscopic traffic flow model inspired on the concepts of hydrodynamics. The LWR model, also known as kinematic wave model, establishes a relation between speed, density and flow of traffic by treating it as a fluid. Considering these rules, Daganzo created a model, discrete in time and space, that could be easily coded and used to describe complicated networks. Another advantage of his model is that it converges rapidly and circumvents LWR's problem of multiple solutions (Ansorge 1990).

A brief introduction to the CTM will be presented here and a complementary description may be seen at Appendix A.2 Cell Transmission Model. The CTM implementation used is the same as in Pohlmann 2010.

The CTM divides a road stretch in the so called cells, where each cell's length, l_i^{cell}, is made big enough so that a vehicle driving through it at free flow speed, v_i^{free}, would

need one time interval unit to reach the next cell:

$$l_i^{\text{cell}} = v_i^{\text{free}} \Delta t \tag{5.15}$$

where Δt is the time difference between each update of the model. The concatenation of the cells make up for the representation of any road segment, as shown in Figure 5.2, and they may or may not represent multiple lane roads, as long as the parameters of each cell are accordingly set.

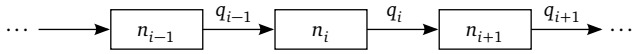


Figure 5.2: CTM cells

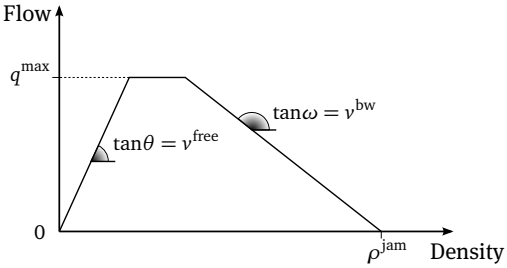


Figure 5.3: Fundamental diagram of the CTM

Figure 5.3 depicts the relation of flow and density in each cell, which is an approximation of the LWR model and governed by:

$$q_i(k) = \mathbf{min}\left\{ n_i(k), q_i^{\max}, \frac{|v^{\text{bw}}|}{v^{\text{free}}} \left(n_{i+1}^{\max} - n_{i+1}(k) \right) \right\} \tag{5.16}$$

where:

- $q_i(k)$ number of vehicles leaving cell i at interval k, [veh]
- $\mathbf{min}\{\ldots\}$ function that returns the minimum value among the elements of the set passed as parameter
- $n_i(k)$ number of vehicles inside cell i at the beginning of interval k, [veh]
- $q_i^{\max}$ maximum number of vehicles that can leave cell i in one time unit interval, [veh]
- $n_{i+1}^{\max}$ maximum number of vehicles that the downstream cell $i+1$ can hold, [veh]

Note that q_i is given in [veh] not in [veh/s] because it actually represents the amount of vehicles that will leave cell i during the time interval k, that lasts Δt seconds. After q_i has been calculated for all cells in the network, at the update interval k, then the amount of vehicles that will be inside each cell at the end of this interval is calculated:

$$n_i(k+1) = n_i(k) + q_{i-1}(k) - q_i(k) \tag{5.17}$$

and this will be the value used on the next update interval at $k+1$.

For the case of representing a signalized traffic network, the traffic signals are modeled by simply setting to zero the outflow q of the cell that coincides with the position of the signal, whenever a red (or yellow) light is in progress. For a truthful mirroring of the signal plans, the time interval, Δt, between each model update is set to one second.

In the context of the queue estimator/predictor presented here, it is also important to highlight the methods used to read and set queues in the CTM. All the signalized links in the network are represented by a defined set of consecutive cells, where the first cell of the set emulates the traffic signal. As seen on Section 3.4.1 Traffic Flow Model the queue length considered by the Green Control Module is the size of the queue at the instant where the according traffic signal changes from red to green.

The method for reading a queue length is really simple. Given the above mentioned time instant, an iterative process is carried out. Starting with the first cell belonging to the link , e.g. cell i, the amount of vehicles inside, n_i, is compared to the maximum amount of vehicles the cell can hold , $n_i^{\max}$. In case they are the same, a counter, previously zeroed, is incremented with $n_i^{\max}$. Then, the next cell, $i-1$, is queried, and so on. When the number of vehicles inside the cell is smaller than its capacity, the counter is incremented with this value and the iteration stops. The resulting number of vehicles gathered by the counter is the assumed queue length (Algorithm 5.1 Queue Scan).

As explained in Section 4.1.2 Offset Module, it is also interesting to have an estimate of the downstream queue actually "seen" by each of the traffic streams associated to the offsets being considered, which means the present state of the downstream queue when the upstream converging links' traffic signals turn from red to green. In this case, it is probable that the queue has already started the discharging process, and the first cells will have less vehicles inside than their capacity. The counter gets incremented by these cell's maximum capacity until it encounters a cell actually holding its maximum number of vehicles. After this point, the same approach as earlier is implemented. In case the counter reaches the last cell in the link, and it is not full, the queue is considered zero (Algorithm 5.2 Queue Scan for the Offset Control Module).

Because the Unscented Kalman Filter requires the ability to apply different values

Algorithm 5.1: Queue Scan

```
N : link's number of cells;
i ← N;
queue ← 0.0;
while i > 0 do
    if n_i = n_i^max then
        queue ← queue + n_i^max;
    else
        queue ← queue + n_i;
        break;
    i ← i − 1;
```

Algorithm 5.2: Queue Scan for the Offset Control Module

```
N : link's number of cells;
i ← N;
queue ← 0.0;
while n_i < n_i^max do
    if i = 1 then
        queue ← 0.0;
        i ← 0;
        break;
    else
        queue ← queue + n_i^max;
    i ← i − 1;
while i > 0 do
    if n_i = n_i^max then
        queue ← queue + n_i^max;
    else
        queue ← queue + n_i;
        break;
    i ← i − 1;
```

Algorithm 5.3: Queue Transfer to CTM

N : link's number of cells;
$i \leftarrow N$;
while $i > 0$ **do**
    if $queue > n_i^{\max}$ **then**
        $n_i \leftarrow n_i^{\max}$;
        $queue \leftarrow queue - n_i^{\max}$;
    else
        $n_i \leftarrow queue$;
        $queue \leftarrow 0$;
    $i \leftarrow i - 1$;

to the system variables, queues must also be set in the CTM. The procedure is quite similar, starting with the first cell in the link, a counter loaded with the desired queue length is decremented by each cell's maximum holding capacity. When the remaining value is smaller than the holding capacity, then this value is given to the cell as the amount of vehicles inside. For the remaining cells upstream the link, the amount of vehicles is set to zero (Algorithm 5.3 Queue Transfer to CTM).

5.3 Unscented Kalman Filter

The Kalman Filter (KF), proposed in Kalman 1960, 1963; Kalman and Bucy 1961, is a linear least-squares estimation technique, or simply linear filtering method, that was conceived to improve the quality of the measurements of a system, which characterize its current state. In a control system, the correct identification of the process' state is of vital importance for its effectiveness. As is usually the case, the measurements are subject to noise and sometimes the desired quantity/dimension is not even physically measurable and/or available. Following the work of Wiener 1949, which successfully related statistical properties to signal and unwanted noise in the communications field (Kailath 1974), Kalman derived an recursive algorithm capable of rejecting part of the noise and outputting a better estimate of the involved variables. Given its recursive characteristic, it may be applied online conferring the technique great usefulness. The present noise is assumed to be Gaussian random, i.e. with an associated probability density function which describes the range of the noise with a mean value and a standard deviation. This means that the measured value of a given state, which incorporates this noise, will also be Gaussian random, and may be characterized by a mean value with a standard deviation. The main idea of the Kalman Filter is to perform a propagation in time of the states of the system from the last iteration, and to fuse them with the available, and current, measured values, which

yields a weighted result. This process takes into consideration the expected statistical distributions of the noises, states, and their relations (their mean values and covariances), and calculates the final estimates by minimizing the expected squared errors between the real values and their statistical estimates (Kalman 1960). The implementation of the KF is described in Appendix A.3 Standard Kalman Filter, and the reader may consult the referenced work in order to gain access to a deeper understanding of the underlying theory behind it.

The Kalman Filter uses, as basis, the statistical properties of Gaussian linear processes. Its use with nonlinear systems is not possible because there is no general analytical expression that describes the probability density function of such systems, which means that, without any prior transformations, it is not possible to propagate in time the necessary mean values and covariance matrices that are used in its calculations. A common solution in these cases, called Extended Kalman Filter (EKF), proposes the linearization of the system's model so that the usual Kalman Filter may be applied (Anderson and Moore 2012). But, this alternative may lead to poor results depending on the original model and point of operation. Following this path, Tampère and Immers 2007 suggested the linearization of the CTM for their traffic state estimator.

The already mentioned works in Jabari and Liu 2012, 2013 consist of a stochastic traffic flow model that retains the characteristics of a CTM, and its Gaussian approximation, characterized by a deterministic mean and covariance dynamics, that enables the use of the standard Kalman Filter for estimating traffic states. Their work has been developed in parallel with the present one, but, even though they showed really good results, it is still not clear if it may be applied for network wide online queue estimation, given time constraints. Anyhow, the influence of the network dynamics and upstream queues on the formation of downstream queues does not seem to play a key role on their solution, apart from needing high resolution data for its operation.

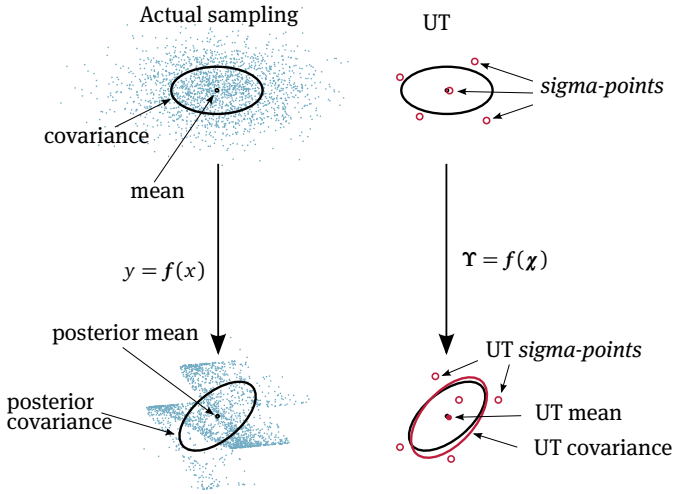


Figure 5.4: Unscented Transformation (adapted from Wan and van der Merwe 2001)

Proposed in Julier et al. 1995; Julier and Uhlmann 1997, the Unscented Kalman Filter (UKF) does not require the linearization of the model, and the nonlinear function that governs the dynamics of the system is used directly. In order to capture the future means and covariance matrix of the model, the UKF evaluates a pre-specified number of possible/probable operation points (proportional to the number of state variables), the so called *sigma-points*. They retain together the expected current mean and covariance values, which, after being processed by the nonlinear model in a time step, also represent the future mean and covariance values of the system. This process is called Unscented Transformation (UT) and is depicted in Figure 5.4. The UKF resembles the Particle Filter (Gordon et al. 1993), but requires a much smaller set of points, which may be of great advantage for time critical applications. This brief concept of the *sigma-points* presented, and how they are used, is enough for understanding their use in the present queue estimator, therefore, its implementation is only described in Appendix A.4 Unscented Kalman Filter.

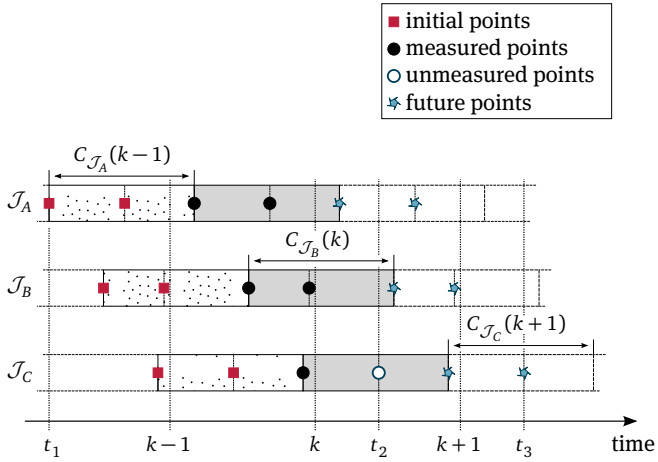


Figure 5.5: Queue points at time k (source: DANTAS and FRIEDRICH 2013)

The use of the UKF in the present work made not only the use of the CTM possible, but also allowed the treatment of the CTM as a black-box model, as if there was an intermediary layer separating the real nonlinear model and the model seen by the Kalman estimator. This decoupling also enables the Kalman Filter to operate in a different time frame/frequency compared to the underlying Cell Transmission Model. The queues are to be estimated for specific time instants, which correspond to the time where the associated traffic signal changes from red to green. Depicted in Figure 5.5, the highlighted points correspond to the beginning of the stages of the signal plans of three successive intersections, during three consecutive cycles. These points mark exactly the time instants where the traffic signals turn from red to green, thus, the time instants for the estimation of the queues. The UKF operates at the same pace of the traffic control system presented in this work, i.e. in a cycle-by-cycle basis. This means that, at every control interval k, the queues of all the links of this period must be estimated, even though their exact time of evaluation is different from one another. The result is that the queue estimator processes the evolution of the queues from the last control interval, $k-1$, until the current one, k, so that the network dynamics are taken into consideration. A time update in the UKF, from $k-1$ to k, corresponds to many more time updates in the CTM: from t_1 until t_2, since, in order to precisely mimic the signal plans, its time resolution is equal to one second, different from the usual 60–120 seconds of the traffic control system. The "initial points" are the time instants that represent the queues that took place during the time interval $k-1$. For the UKF, these points correspond to the expected previous state of the system. Analogously, the "measured points" and "unmeasured points" are related to the current time inter-

val k. Because of the fact that the control algorithm is always called before the end of the first signal plan of the current time interval, there may be some local queue estimations that will not be available. Considering that k is the current time interval in Figure 5.5, the local queue estimation related to the second stage of $\mathcal{J}_C$ is taken as an unmeasured point. The UKF will then estimate the current queue sizes of time interval k based on the expected previous queue sizes of time interval $k-1$ and the measured queues of time interval k. The set of queue values of time interval $k-1$ represents the mean values of the probable queues, and, as explained earlier, based on the expected noise inherent to the system, another sets of queues will be generated, the *sigma-points*, whose dispersion represent the expected covariance between the states of the system. Each *sigma-point* is a set of queue values which will be applied to the CTM. These queues are introduced at the "initial points", using the technique described in Section 5.2 Cell Transmission Model, during a simulation run of the CTM between t_1 and t_2. The posterior queues collected as a result of each *sigma-point* simulation, that represent possible queue sets of time interval k, are later on processed by the UKF with the real measured queue values resulting in a final estimation of all the queues.

After the estimation of the queues through the filtering step, a simple prediction step is carried out. The estimated queues belong to interval k, but the control strategy will be calculating the necessary control actions for the interval $k+1$. Therefore, a final simulation of the CTM is performed, from time instant t_1 until t_3, where the estimated queues are applied to the CTM at the measured and unmeasured points. Since the future signal plans are not yet available, the latest ones are replicated for time interval $k+1$ and the resulting queues, depicted as "future points", are collected from the CTM.

6 | Evaluation

6.1 System Setup

6.1.1 Test Network

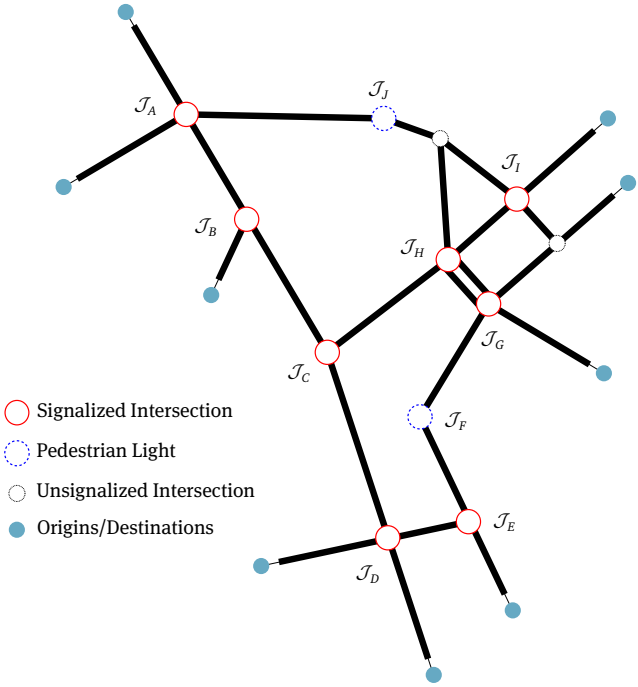


Figure 6.1: List network (source: Dantas and Friedrich 2013)

In order to test the current developments, the same network used in POHLMANN 2010 was chosen. Not only the work of collecting and adjusting the available data for the use in a simulation environment was ready, but also the possibility of comparing both strategies were key for the decision. The network in subject is part of the traffic network of the List[1] district in Hanover, Germany. As shown in Figure 6.1, it consists of 10 signalized intersections, being two of them pedestrian signals with the pedestrian stages with fixed lengths. Actually, it is a striped down version of the real network, where secondary streets were left out.

Almost all of the represented links are two-way streets, with one lane for each direction. The exceptions are the two links between $\mathcal{J}_G$ and $\mathcal{J}_H$, where each is a one-way street with two lanes. Totaling 55 monitored links, they comprise the additional 17 left-turn and 5 right-turn lanes, which measure, in average, 35 meters in length, and are distributed across the network. The speed limit of the network is 50 km/h, with the exception of a couple of links with 30 km/h.

6.1.2 Simulation Environment

For the evaluation process, AIMSUN[2] (version 6.1.5), a renowned microscopic traffic simulator, was used. AIMSUN has also been used in Pohlmann's work, and the existing simulation model of the mentioned network was taken. In order to build a more realistic traffic simulation, Pohlmann gathered real traffic data available from local loop detectors, that were provided by the city of Hanover. These data were complemented by another six radar detectors that were installed in strategic positions for the same day, helping to better assess the traffic demand. With this information he managed to recreate a meaningful traffic scenario for the period between 6 AM to 8 PM. For each 15-minute period, a different Origin-Destination Matrix has been created, and together they reproduce a realistic traffic demand profile, covering almost an entire day, for this specific network.

[1] Represented in Google Maps: https://maps.google.com/maps?q=List,+Hanover,+Germany&hl=en&ll=52.389181,9.75086&spn=0.008879,0.018625&sll=37.0625,-95.677068&sspn=46.898798,76.289063&oq=list+ha&hnear=List,+Hannover,+Germany&t=m&z=16 (visited on 10/09/2013)

[2] A traffic simulation program from TSS – Transport Simulation Systems, http://www.aimsun.com/wp/ (visited on 10/09/2013)

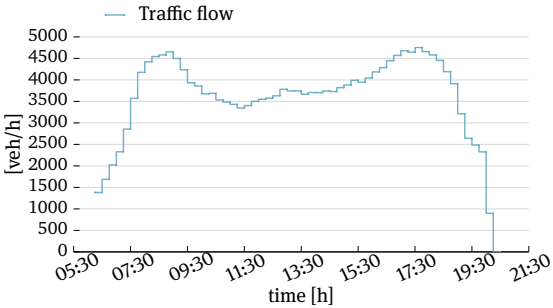


Figure 6.2: Traffic flow

Figure 6.2 shows the resulting average traffic flow profile for the whole network from 10 different simulation replications. For the present work, an additional 30-minute interval was added to the end of the simulation period. Within this last time interval no vehicles are generated. This allows all vehicles inside to leave the network before the simulation ends, guaranteeing that the last vehicle trips are integrally completed and accounted for in the final results. Another modification to the existing simulation scenario was the lengthening of the peripheral incoming links, which assures that no traffic congestion will influence the generation of vehicles by the simulator, as may be the case. Whenever an incoming link is completely full of vehicles, the simulator delays the entrance of newly generated vehicles, and may also discard them. This results in less vehicles being processed by the replication, turning the amount of vehicles generated by the exact same replication unpredictable and variable depending on the control strategy being used. Complementing the above measures, all turning lanes were transformed into separate links. Before, vehicles were allowed to switch to the turning lane at any point of its length, and now they are forced to to enter the turning lane at the bifurcation of the upstream single lane. This avoids that unsuccessful vehicles trying to jump in queues end up missing the turning and travelling routes that they are not supposed to. All these precautions guarantee that the exact same routes will be traveled by the exact same number of vehicles for every run of the same replication, independent from eventual adverse traffic conditions or from the control strategy being enforced.

Both of the local queue estimation strategies tested, Mück's and Vigos's (presented in Sections 5.1.1 and 5.1.2), expect the positioning of the inductance loop detectors at a given distance from the stop line. The original detector positions present in Pohlmann's simulation model did not apply to these expected distances. Therefore, new detectors were created for each of the monitored links in the network. They were all positioned as far apart as possible from the stop line, i.e. right after the bifurcation of the lane as seen in Figure 6.3, which means at an average distance of 30 meters from

the stop line. Mück's estimator expects the positioning of detectors at a distance usually chosen by actuated traffic control solutions that use the headway between vehicles as input measurement. This distance is equivalent to 1/10–1/5 the expected queue length during peak hours (Mück 2002a). In Vigos's estimator, it is expected that the detectors are placed at the middle of the link. The new detectors and their positions were chosen trying to find a compromise between these two guidelines and where they would still be capable of capturing individual streams of traffic. Therefore, they are not really optimal to either alternative but attain a better position than earlier, where they were too close to the stop-line.

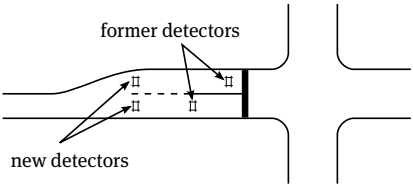


Figure 6.3: New detector positioning

6.1.3 External Traffic Control Module

The developments of the current work were applied to AIMSUN through its Application Programming Interface (API), which allows all sorts of external control over the simulation. Through this API all traffic signals are directly commanded and the measurements of the emulated detectors are read.

All the coding necessary used Python[3] (version 2.7) as programming language, but some of it was also implemented in Java[4] (version 7). The actual implementation of the CTM, used in the Queue Estimator, was completely taken from Pohlmann 2010. The code is written in Java, and therefore almost all of the Queue Estimator module was also written in Java and connected to the main code through a Pyro[5]/Jython[6] bridge.

Another third-party software used was the optimization solver CPLEX[7] (version 12.5.1) from IBM. It was used for solving the quadratic minimization problem that is part of the Green Control Module of the system.

Websites last visited on 10/09/2013
[3] http://www.python.org
[4] http://www.java.com
[5] https://pypi.python.org/pypi/Pyro4
[6] http://www.jython.org/
[7] http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/

6.1.3.1 CTM

The queue predictor is also part of the external control module. Given the modifications made to the simulation model, it was also necessary to recalibrate the CTM model used in the queue estimator. In this process many left and right turn paths had the free flow speed, v_z^{free}, reduced to 25–30km/h. All v^{bw} were set to 18km/h. Since there is no estimation of the turning rates, each value was set to the mean value spanning the whole day of the described scenario.

6.1.3.2 Green Control Module

Just like the CTM used in the queue predictor, the traffic flow model used as basis in the Green Control Module also requires the definition of turning rates, which describes the distribution of traffic throughout the network. As mentioned above, these values were considered constant and equal to the average values expected during a whole day operation of the given scenario.

In the minimization of the quadratic problem, a horizon of 5 cycles ($N_{\text{p}} = 5$) was used for all the evaluations.

6.2 Queue Predictor Evaluation

Three different statistical indicators were used to assess the performance of the Queue Predictor. The first one is the Correlation Coefficient, r:

$$r = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}} \tag{6.1}$$

where:

- N number of sample pairs of the two variables being evaluated
- x_i the actual value of a given variable from the i^{th} sample
- y_i the estimated value of a given variable from the i^{th} sample
- $\bar{x}$ the arithmetic mean value from N samples of x
- $\bar{y}$ the arithmetic mean value from N samples of y

It is a measure of linear correlation between two variables, i.e. their linear dependence. It helps to analyse how well the estimated value is following the evolution of the real value. The output of Equation 6.1 is dimensionless and may be in the range$[-1, 1]$, where the extremes indicate total correlation, negative or positive, and 0 indicates no correlation.

The second indicator is the Root Mean Square Error, RMSE:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(x_i - y_i\right)^2} \tag{6.2}$$

whose output shows, in terms of the same unit being measured, the difference between the estimated and real values, i.e. the closer to zero the better.

The last one is the Relative Mean Square Error, ReMSE:

$$\text{ReMSE} = \frac{\sqrt{N\sum_{i=1}^{N}\left(x_i - y_i\right)^2}}{\sum_{i=1}^{N} x_i} \tag{6.3}$$

which is dimensionless, and accounts for the variable's magnitude. Depending on the range of the measured values, the error between the estimated value and the real one may not be that great comparing its magnitude and the magnitude of the variable. The given formula was chosen because null values may also be used without raising *division-by-zero* errors, which usually occurs when dealing with queue lengths.

The key feature of the Queue Predictor presented is the use of the network's dynamics, i.e. the interaction between the network's consecutive links, in order to improve the queue estimation. Therefore, an isolated link scenario would not make sense for its evaluation, and because of that the complete List Network was used. The results are based on the average results collected from all the 55 links, during 10 different replication runs of the scenario described in Section 6.1.

The assumed real queue lengths are the ones collected from the simulation during the switch of the traffic lights from red to green. As mentioned on the previous Section, almost all links are single lane road stretches that divide near the intersection in order to derive a turning lane. Whenever the queues stretch upstream into the single lane, a simple algorithm is used in order to separate the vehicles that are heading to the turning lane. The number of vehicles stopped in the single lane is simply multiplied by the average turning rate expected from each location, and each percentage is associated to the according downstream lane as additional queue.

In real conditions, the detector measurements are constantly under the influence of noise, which must be accounted for. The measurement errors may come from defective sensors; from detectors being triggered from vehicles passing by a neighbouring lane; the speed at which the vehicles cross the detector; or even the simple fact that vehicles have different lengths and produce different readings, specially for occupancy measurements. Since some of these errors cannot be emulated by the traffic simulator, a straightforward modification is employed in order to confront the Queue Predictor with a more realistic scenario. All detector data from the simulator is preprocessed before being delivered to the traffic control system. An independent multiplicative noise is created for each variable being measured. This noise is actually a set of random numbers which are generated through a mathematical library, in the programming language being used. This set is generated from a specified seed number, and have a Normal distribution with mean $\mu = 1.0$, and standard deviation $\sigma = 0.2$, $\mathcal{N}(\mu, \sigma^2)$, as chosen in the current work. For every new reading, one value is drawn from the associated noise vector and is multiplied to the reading value, resulting in a measurement with 20% multiplicative noise. For the sake of reproducibility, the seed of each noise is a combination of the detector's and the replication's identification numbers. This guarantees that the noise will be the same for every run of the same replication.

The evaluation process consisted of three steps. In the first two, the proposed Queue Predictor was applied to the whole day simulation scenario presented in the previous Section, with fixed time signal plans. The purpose was to eliminate the varying signal plans on every new cycle, as a result of the online traffic control, which could have masked the expected output of the queue predictor. Do not forget that the prediction step of the algorithm simply apply the last signal plan used. In the last test, the TUC strategy was used and the queue predictor has been evaluated against its intended use.

In the first battery of tests, only the estimation capability was evaluated. For this part, the last step of the proposed Queue Predictor was shut down, so that only the estimated queues were evaluated, i.e. there was no prediction involved. The results are summarized in Tables 6.1 and 6.2, and show how the Queue Estimator, with the UKF attached, performs against each of the two local queue estimation techniques taken as reference/source of "pre-measurement". The alternative named Mück+UKF uses Mück's queues as raw-measurement for the UKF, and Vigos+UKF uses Vigos's queues.

The concept of using the network dynamics for improving the queue estimation might have been obfuscated by the performance of the UKF itself in rejecting noisy measurements. As proof of concept, the noise being injected in the detectors' readings was completely removed, and a preliminary test was carried out by only evaluating the Mück's related variants. Comparing the results in Table 6.1 and 6.2, it is clear that

the algorithm is not only capable of rejecting noise but it really improves the overall measurement errors.

Alternative	Mean RMSE [veh]	Mean ReMSE [-]	r ×100[-]
Mück	2.5316	1.2859	33.78
Mück+UKF	2.1664	0.8918	38.64

Table 6.1: Fixed time estimation-only results (no noise)

Alternative	Mean RMSE [veh]	Mean ReMSE [-]	r ×100[-]
Mück	2.6018	1.3142	31.50
Mück+UKF	2.2233	0.9063	36.22
Vigos	2.1320	0.9052	60.95
Vigos+UKF	2.0684	0.8498	53.00

Table 6.2: Fixed time estimation-only results (with 20% multiplicative noise)

Table 6.2 shows that Vigos's local queue estimator performs better than Mück's. In both cases, the UKF-based queue estimator was capable of reducing the RMSE and the ReMSE indicators. The exception is only seen in terms of the correlation coefficient that decreased with the use of the proposed queue estimator in Vigos+UKF. This actually exposes a limitation of the algorithm that ends up smoothing the output from a certain level, reducing the correlation coefficient while still reducing the error. A brief visualization on how these three indicators respond to different scenarios may be seen in Appendix A.5 Performance Comparison of the Statistical Indicators Used.

At the second test batch, the prediction step of the algorithm was applied and a new configuration of the queue predictor have also been tested. In order to really confirm the benefit brought by the inclusion of the UKF in the queue estimator algorithm, another similar, but much simpler, queue predictor was used. Depicted as Mück+CTM and Vigos+CTM, these two variants take the estimated queue values from the respective local queue estimators and apply them directly to the CTM model for a prediction run. This bypasses the UKF step. The other two variants, Mück+UKF$^{\text{pred}}$ and Vigos+UKF$^{\text{pred}}$, represent the use of the queue predictor as it was intended to.

Alternative	Mean RMSE [veh]	$5 \leqslant$ RMSE < 10 [%]	RMSE > 10 [%]	Mean ReMSE [-]	r [-]
Mück	2.7375	10.91	23.96	1.3652	19.99
Mück+CTM	2.6136	12.11	20.87	1.1606	27.20
Mück+UKF$^{\text{pred}}$	2.3922	10.03	19.41	0.9752	24.96
Vigos	2.5950	10.89	23.60	1.1006	25.71
Vigos+CTM	2.3273	10.63	16.45	0.9925	30.75
Vigos+UKF$^{\text{pred}}$	2.3436	9.86	19.16	0.9443	27.40
Real+CTM	2.8885	10.15	22.03	1.2244	29.20
Real+UKF$^{\text{pred}}$	2.2572	8.73	16.73	0.9269	29.44

Table 6.3: Fixed Time prediction results (with 20% multiplicative noise)

Additionally, the real queues, modified with the mentioned noise, were used in both of the queue predictor variants (Real+CTM and Real+UKF$^{\text{pred}}$). The intention of this last modification was to verify how the error introduced by the local queue estimation affected the final results. The results are shown in Table 6.3, where two additional statistics are introduced. One of them indicates the percentage of all measurements that had a RMSE between 5–10 vehicles, and the other, the percentage of them that had a RMSE bigger than 10 vehicles.

This second set of tests shows that the simple prediction step performed really improves the measurements. Note that, in this evaluation, the queues predicted in a previous cycle are being compared to the queues that only occurred on the next cycle. What is interesting to point out is that, for the case of using the real queues in conjunction with the simple CTM prediction step, Real+CTM, there was actually a degradation of the results. This could be explained by arguing that both local queue estimation techniques used, Mück's and Vigos's, even without the UKF, help by filtering out some of the noise added. Apart from that, the performances of Mück's and Vigos's local queue estimators do not seem to impact the final results as much as the prediction error itself. As the overall results show, the correlation coefficients are low, ranging from around .25–.30, and the percentage of the measurements with a RMSE greater than 5 vehicles are quite high, reaching up to around 30%, including the Real+UKF$^{\text{pred}}$ variant. This means that even with the improvements brought by the prediction step, there is still a good room for improvement left. Another result that gets the attention is the correlation coefficient of the Vigos+CTM variant, which is higher than the ones produced by using the real queues in Real+CTM and Real+UKF$^{\text{pred}}$. Also intriguing is the fact that even without a real filtering step, Vigos+CTM had an overall better performance than Vigos+UKF$^{\text{pred}}$, even though the ReMSE was lower for the latter one. These results indicate some evidence that the synergy between each module of the queue predictor may also play an important role on its final results, discarding the initial impression that the previous tests presented, where Vigos+CTM seemed to

have performed better.

With the first tests, it was clear that the UKF algorithm was capable of improving the queue measurements. But, as the case of using the Vigos+CTM alternative showed, it could perhaps not be that much more beneficial in comparison to a simpler approach. In order to really measure the impact of the queue estimators presented, the final set of tests were performed with the TUC strategy controlling the signal plans, and the traffic performance indexes: Average Delay; and Fuel Consumption were evaluated.

Alternative	Average Delay [s/km]	Fuel Consumption [l]
Mück+CTM	21.7414	12468.3
Mück+UKF^{pred}	20.6256	12343.5
Vigos+CTM	21.3965	12417.5
Vigos+UKF^{pred}	20.5560	12345.6
Real+UKF^{pred}	20.7251	12334.5

Table 6.4: TUC results (with 20% multiplicative noise)

As Table 6.4 shows, the UKF-based queue predictors were able to deliver better results than the simpler Mück+CTM and Vigos+CTM alternatives. In terms of Average Delay, they were even better than Real+UKF^{pred}, which may come as a surprise. But, the Fuel Consumption values point out that, indeed, Real+UKF^{pred} had a slightly better performance. This may be explained to the more accurate estimation of the queue lengths, which means that the offsets being calculated are also closer to their "ideal" values, therefore reducing stops and consequently fuel consumption. The higher average delay in Real+UKF^{pred} indicates that the streams not being served by an offset policy must wait a little longer highlighting the fact that a better offset for a particular stream, in general terms, implies a worse offset for a competing stream. The final results also show that the ReMSE indicator had the closest relation to the final output performance of the queue estimator.

Weighting the results from all the tests, the alternative Vigos+UKF^{pred} was chosen as the best one and has been used throughout the remaining work.

6.3 Cycle Alternatives Evaluation

In Chapter 3 TUC, two options for controlling cycle lengths were presented. They will be referred to as Webster, for the one based on Webster's formula, and SAT for the

other. The original control scheme was left untested since the objective was to have a better means of comparison with Pohlmann's ATCS (Pohlmann 2010), taken as reference later on. The tests carried out here used the proposed Selfish Strategy presented in Section 4.1.2.2 of Chapter 4 Proposed Modifications to TUC. The input data used was the average vehicle count from the last 3 cycles converted to vehicle demand by simply dividing the value by the according period. The best distance found between each update of the common cycle length was once every 7 cycles.

For the evaluation process, an additional scenario has been tested. The original traffic demands were increased 15% in order to test the control technique in a more challenging situation. This increase has led the peak hours into a congested traffic state, where wrong control measures may be better spotted. The reference cycles are limited by the Cycle Control Module to be between 54 and 110 seconds, and the cycles are allowed to vary between 49 and 140 seconds by the Green Control Module. The results are presented in Table 6.5.

Demand	Cycle Type	Average Delay [s/km]	Fuel Consumption [l]
Normal	Webster	19.9987	12291.0
	SAT	20.1905	12446.9
115%	Webster	23.9631	14608.0
	SAT	26.3477	15072.9

Table 6.5: Simulation results (with 20% multiplicative noise)

In the original/normal scenario, the alternative using Webster cycles had a slightly better performance compared to the Saturation Based cycles. This happened despite the relative big differences in cycle lengths, specially during light traffic, as shown in Figure 6.4, where the average cycle lengths from each alternative are aggregated in 15-minute intervals. Actually, analysing the average delays in Figure 6.5, it is clear that the smaller cycles during light traffic conditions were slightly beneficial for the operation of the network. But, the average 5 seconds longer cycles during peak hours guaranteed an overall better performance for the Webster alternative.

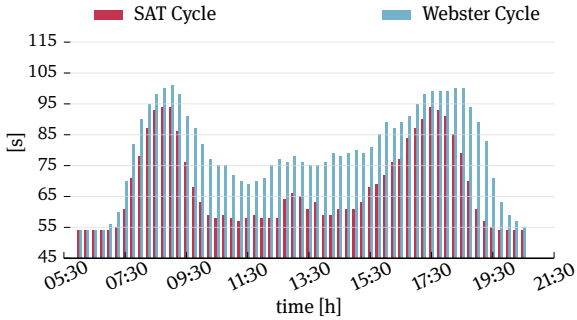


Figure 6.4: Webster and SAT reference cycles

Figure 6.6 depicts the average cycles for the scenario with increased traffic demands. Crossing the information in the figure with the results in Table 6.5, one would expect that the greater advantage presented by the Webster based cycle control would come purely from the performance of the system between around 10AM until 16PM, since the cycles in the peak hours are almost the same. But, as depicted in Figure 6.7, a great amount of the difference comes from the second peak hour. This shows that the not so favorable conditions, or lack of playroom, before the onset of the congested period has compromised the system's performance for the rest of it's length.

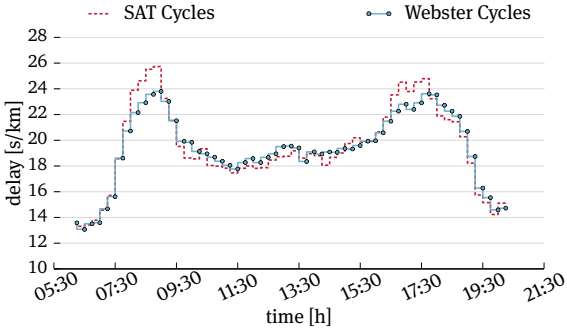


Figure 6.5: Webster vs. SAT delays

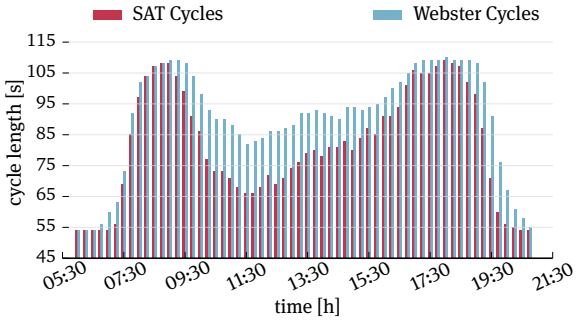


Figure 6.6: Webster and SAT reference cycles – 115%

These results corroborate with the ones from Pohlmann's solution, as presented in Appendix A.6 Tests with Pohlmann's ATCS. The Saturation Based cycles do not seem to be adequate to be used in an adaptive traffic control system. The cycles are too short and less forgiving in errors committed by the control strategy, or in possible stronger traffic oscillations.

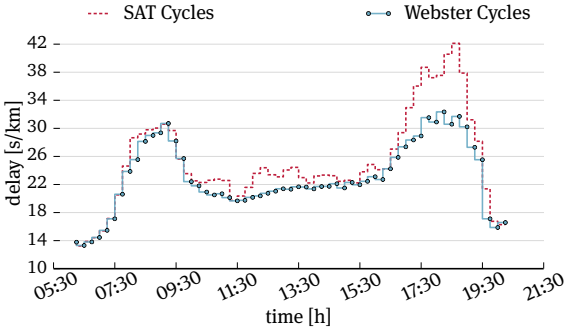


Figure 6.7: Webster vs. SAT delays – 115%

6.4 Offset Alternatives Evaluation

The main focus of the present work was to improve the way that the TUC strategy implements offset control. The two main goals were to avoid abrupt cycle changes during transition cycles, and enable TUC to handle offsets between secondary stages, which also allows the implementation of offsets in a meshed network.

As presented in Chapter 4 Proposed Modifications to TUC, two alternatives were developed. The first one, which will be referred to as *Proposed*, represents the original intention of this work, where the Transition Cycles used to implement new offsets are calculated along with the green time durations by the Green Control Module. And the second one, referred to as *Legacy*, serves as refutation of the former by simply limiting the length of the Transition Cycles to 80% or 120% of the current reference cycle. In this case, the Transition Cycles are, as originally, dictated by the Offset Control Module and their values are just imposed to the Green Control Module.

The *Legacy* alternative has two modifications compared to the original TUC strategy. One is the ability to implement secondary offsets, and the other, the implementation of smoother transition cycles, limited to ±20% of the reference cycle. For better evaluating the separate impact of each of these modifications, the *Legacy* alternative has been tested with two options for transition cycles. One of them, referred to as *Abrupt*, implements the same one-cycle transition cycles as TUC does for implementing new offsets. Comparing this alternative with TUC, it is possible to assess the gains uniquely brought by the capability of considering secondary offsets. And the second one, referred to as *Smooth*, implements both modifications. Comparing the *Legacy Abrupt* and the *Legacy Smooth* alternatives, the contribution brought by the use of smoother transition cycles is highlighted.

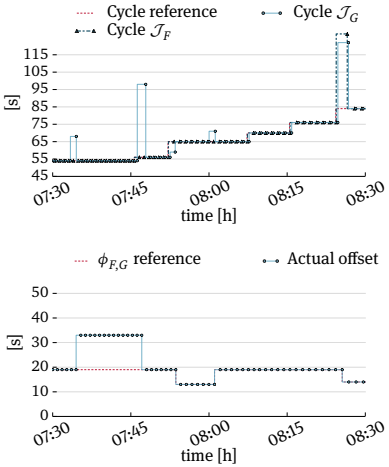
Besides the main objectives already mentioned, and taking advantage of the open opportunity, the current investigation also dealt with the determination of the reference offsets. This effort is summarized in the two additional offset strategies presented in Chapter 4, Sections 4.1.2.3 Democratic Strategy and 4.1.2.4 Partially Democratic Strategy, which offer an unconventional form of determining the offsets.

In order to verify whether the *Proposed* alternative was really capable of following the new offset and cycle references without the abrupt cycle changes, common to the original TUC strategy, a simple analysis of the progression of the cycles and offsets has been carried out. As shown in Figure 6.8, a one hour sample of the progression of offsets and cycles from intersections $\mathcal{J}_F$ and $\mathcal{J}_G$ has been extracted from one of the replication runs of the test scenario for three similar options available: TUC; Proposed Selfish; and Legacy Selfish Smooth. This specific time period, and involved variables, was chosen because it also portrays a moment where the depicted offset reference, $\phi^{\text{ref}}_{F,G}$, looses priority for the other direction, which would be $\phi^{\text{ref}}_{G,F}$, and therefore a noticeable gap appears between reference and actual offset around the first 15 minutes. This moment, therefore, highlights the existing conflict of objectives that arise during the traffic control.

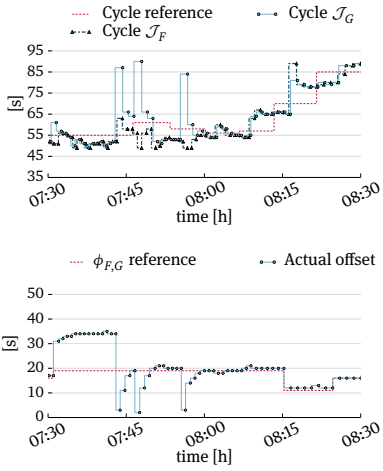
As the Figure 6.8 confirms, the Proposed Selfish alternative reduced indeed the high transition cycles, even though they are still higher than the ones from the Legacy Selfish Smooth alternative. It is also clear that the Proposed alternative allows a much

greater error in cycle following than in offset following, which reflects the difference in the gains used in the cost function of the algorithm. As will be presented next, the apparently unsteady behaviour of the Proposed alternative is not detrimental for the traffic performance. Actually, as previously verified in Section 6.3 Cycle Alternatives Evaluation, the control strategy is capable, at some extent, of absorbing the impact of using non-optimal reference cycle lengths.

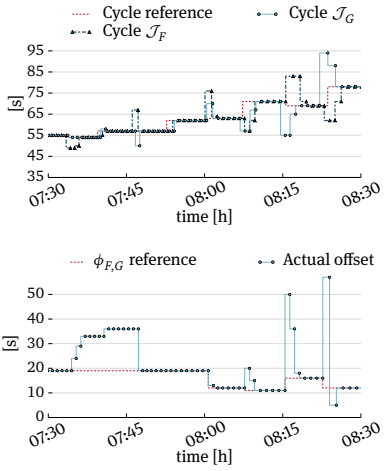
All the possible combinations of offset control mentioned in the beginning of this Section have been tested and the results are summarized in Table 6.6. Even though it was not presented here, all the preliminary tests, with the proposed modifications to the TUC strategy, were performed with the aid of a much simpler virtual network. The results have been documented in Dantas and Friedrich 2012. The virtual test network used consisted of just three intersections and the links' lengths averaged 100 meters in length. Just like in the preliminary tests, the improvement in performance brought by the introduced modifications is quite low in light traffic conditions. In the other hand, the improvements achieved during heavier traffic were not as big as they were in the virtual network. This difference shows that the impact of abrupt changes in cycle length are only really noticeable when there is not enough buffer for accommodating the traffic oscillations. In the case of the presented test network, the links average around 200 meters in length, and therefore are capable of better absorbing eventual traffic disruptions, without letting them spread upstream-wards. The same scenario with 15% increase in overall traffic demand used in Section 6.3 has also been evaluated and the results are grouped in Table 6.7.



(a) TUC



(b) Proposed Selfish



(c) Legacy Selfish Smooth

Figure 6.8: Reference following in intersections $\mathcal{J}_F$ and $\mathcal{J}_G$

Alternative	Average Delay [s/km]	Fuel Consumption [l]
TUC	20.5560	12345.6
Legacy Selfish Abrupt	20.3647	12310.5
Legacy Selfish Smooth	20.1473	12289.9
Legacy Partially Democratic Abrupt	20.5344	12245.8
Legacy Partially Democratic Smooth	20.4789	12239.3
Legacy Democratic Abrupt	20.6394	12275.2
Legacy Democratic Smooth	20.6043	12264.9
Proposed Selfish	19.9987	12291.0
Proposed Partially Democratic	20.2665	12236.7
Proposed Democratic	20.4139	12274.3

Table 6.6: Simulation results (with 20% multiplicative noise)

Looking at Table 6.6 it is possible to draw the first conclusions. Comparing the Original TUC strategy with the Legacy Selfish Abrupt and the Legacy Selfish Smooth alternatives, it is possible to state that both incremental modifications were capable of improving the traffic control performance of TUC. The use of the *Smooth* variant in all *Legacy* options has confirmed the benefit of less abrupt cycle changes during offset maintenance. The *Proposed* alternative has also performed slightly better than the *Legacy* alternative in almost all cases when comparing the compatible alternatives: Proposed Selfish with Legacy Selfish Smooth; Proposed Partially Democratic with Legacy Partially Democratic Smooth; and Proposed Democratic with Legacy Democratic Smooth. Visually inspecting Figures 6.9 and 6.10, as expected, the benefits of the modifications to the original TUC strategy are concentrated in the peak traffic periods, but between the alternatives the differences are evenly distributed.

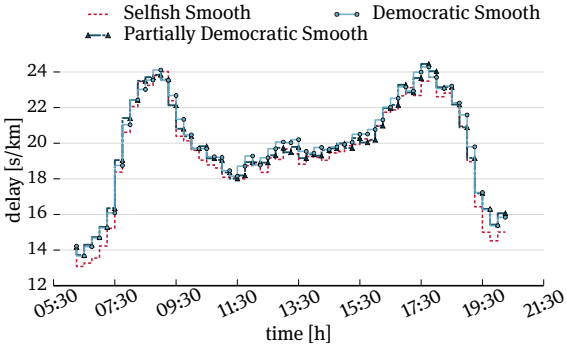


Figure 6.9: Delays for the Legacy variants

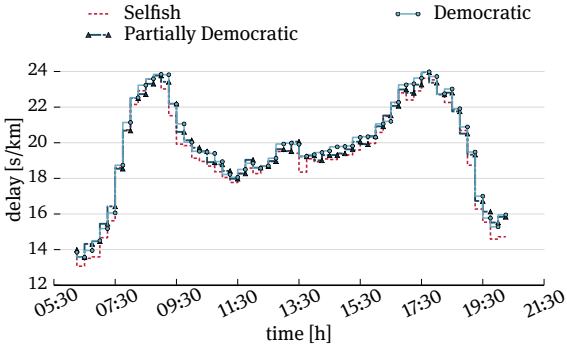


Figure 6.10: Delays for the Proposed variants

The results of the more demanding scenario in Table 6.7 presents a different picture. Even though the improvement over TUC is accentuated, as expected, the Legacy options had, in average, a better output than the Proposed ones. At first glance, and recalling Figure 6.8, one might think that the transition cycles, implemented by the Proposed alternative, may have been kept too big for the given traffic conditions. In order to confirm this assumption, the minimum and maximum cycle constraints, used in the control problem described the set of equations 4.1.2 in Chapter 4, were restrained at each update to not overcome ±20% of the current reference cycle, which forces the Proposed alternative to behave more like the Legacy alternative. The result of this modification is presented in Table 6.7 as *Proposed Selfish (limited cycle changes)*. It shows that the Proposed alternative did not perform worse because of the higher transition cycles, but probably because the reference cycles were not being strictly followed as Figure 6.8 also demonstrates. Either way, the Proposed alternative did not deliver the expected improvement over the much simpler Legacy alternative.

Alternative	Average Delay [s/km]	Fuel Consumption [l]
TUC	24.8173	14753.7
Legacy Selfish Smooth	23.7030	14538.6
Legacy Partially Democratic Smooth	24.1227	14530.1
Legacy Democratic Smooth	24.6415	14589.3
Proposed Selfish	23.9631	14608.0
Proposed Selfish (limited cycle changes)	24.7533	14739.2
Proposed Partially Democratic	23.8618	14496.7
Proposed Democratic	24.6993	14612.0

Table 6.7: Simulation results 115% (with 20% multiplicative noise)

With an overview of these results in mind, it is now possible to address the impact of the unpretentious new methods tested for defining the offset references: Partially Democratic; and Democratic variants. Attaining to TUC's philosophy, these alternatives were conceived with the expectation that abdicating the use of complex and time consuming traffic models, it might have been possible to define simple rules for the the definition of reference offsets that would result in a better traffic operation. As presented in Section 4.1.2 Offset Module, Chapter 4, these methods were based on the observation of the traffic behaviour during a series of simulations tests. As the results show, the Democratic variant proved to be the worst of them but still better than the original TUC. The Partially Democratic, in the other hand, has achieved a better balance between fuel consumption and delay, specially when applied to the Proposed alternative. Despite the small differences between each variant, the results indicate that the assumption made was not completely wrong: which states that the offset which enable the overall reduction of delays is a combination of the ideal offsets (in terms of the formation of green-waves) of each of the competing traffic streams. But, as the next performance comparison will show, the Partially Democratic variant did not come close to reduce the delays as much as it would have been possible to. A more thorough visual analysis of the formation of queues and its dependence to offsets have shown that this approach cannot guarantee that the ideal offset (the one that incurs in less delays) will be reached. This has to do with the fact that a simple change in the offset results in a change of the queues that each traffic stream coming from the upstream intersection "sees". This means that the control system must be constantly adjusting the offsets. Moreover, even in the Selfish variant, the adjustment of the offset to guarantee a green-wave may lead, in some cases, to the increase of the queue lengths. These cases are particular visible when the control system increases the offset value in order to give more time for the dissipation of the standing queue. A few tests have showed that, in some of these cases, by actually decreasing the offset value, the queues retracted and even though the first vehicles of the approaching platoon ended up being forced to decelerate, the progression was maintained and the resulting delays were reduced in comparison to the option of increasing the offset.

Concluding the evaluation, a final set of tests has been carried out. It compares the performance of the modified TUC, proposed in the current work, and the performances of Pohlmann's ATCS and an optimized fixed time alternative. This fixed time alternative is the same that was used in Pohlmann's work as reference case, which is a set of fixed time signal plans that had been prepared with the aid of TRANSYT. For each intersection, two different fixed time signal plans were made, one for the morning period and the other for the afternoon period. Both of them were fine tuned to better respond to the peak traffic hours around 8:30AM and 17:30PM, and feature a 90 seconds cycle length and suitable offsets. The final comparison is presented in Table 6.8 and Figure 6.11. Pohlmann's solution and TUC are two completely different approaches to online traffic control. Trying to minimize the differences, the same method employed

by Pohlmann for deriving the common cycle of the network was used in the current TUC implementation. This effort aimed the reduction of disparities that may derive from using different sets of cycles for a given traffic condition. Unfortunately, because of the completely different treatment applied to the information gathered from the detectors, the values of the resulting cycles being generated by the same method were different. In TUC's implementation, the necessary traffic demands applied in the Webster's formula (see Section 3.2.1 Webster's Cycle Alternative, Chapter 3) are the simple average of the last three readings. In Pohlmann's solution, the traffic demands are aggregated from the last 15 minutes and further processed by the Demand Estimator/Predictor. The outcome is that Pohlmann's set of implemented cycles are up to 20 seconds shorter in the peak periods, nullifying the initial intention. Moreover, the offsets employed in TUC are directed to prioritize green-waves for the most demanding traffic stream, whereas Pohlmann's aims the pure overall reduction of delays. In order to find out how TUC would perform if the offsets were targeted to reduce delays, the offsets from Pohlmann's system were applied to TUC. For each of the ten replication runs, the offset and cycle references selected for each of the 15-minute intervals were accordingly applied during TUC's operation.

Alternative	Average Delay [s/km]	Fuel Consumption [l]
Fixed time	19.0926	11841.1
Pohlmann 15min	17.8852	11947.5
Proposed Selfish	19.9987	12291.0
Proposed with Pohlmann's offsets/cycles	18.5309	12164.1

Table 6.8: Simulation results

Looking at the results, the first point that gets the attention is the fact that TUC's performance (even after the small improvements brought by the current work) was worse than the optimized fixed time alternative. This may be due to the fact that the optimization settings used in TRANSYT were configured to minimize fuel consumption, delays and queues, rather than guarantee the onset of green-waves in predetermined routes. Pohlmann's performance, in the other hand, was capable of delivering better results in almost all periods, with the exception of the last peak hour. Pohlmann's greatest gains are during the light traffic periods, where the smaller cycles being employed seem to potentialize the seen differences. Do not forget that there are only two sets of fixed time signal plans, one for the morning period and the other for the afternoon.

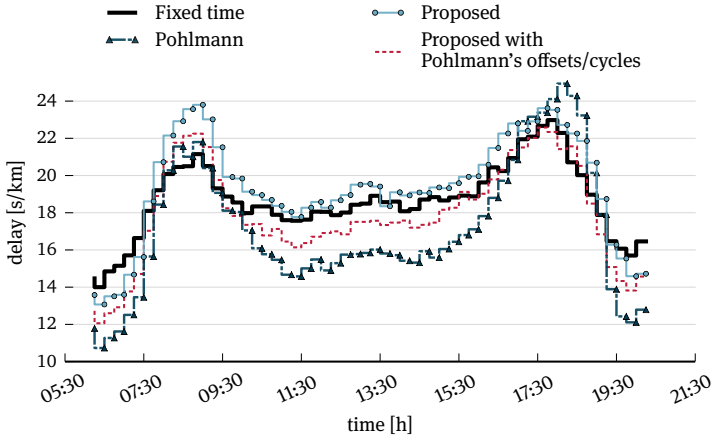


Figure 6.11: Delays comparison

Rather interesting is the performance of TUC when being fed with the offset and cycle references from Pohlmann's solution. Even though the improvement is evident in all periods, a barrier seems to exist during the low traffic periods. An analysis of the proportion of the green times in the cycles has revealed a weak point in TUC's control formulation. In low traffic conditions, the cycles being passed as reference are greater than the sum of green times (including the interstages) claimed by the phases through the evaluation of TUC's traffic model. This means that the exceeding, and available, green time is equally distributed among the stages, distorting the proportion that should have been kept. When comparing the average splits being generated by TUC, and the ones from Pohlmann's, during these conditions, this effect is quite obvious with differences around 15 seconds in some cases. The difference between the fixed time and Pohlmann's may be explained given the much higher cycle lengths and offset configuration during the low traffic periods where there was no adjusted fixed time available. The better fuel consumption values presented by the fixed time alternative when comparing it to Pohlmann's, comes solely from its better performance during the second peak hour, whereas in the rest of the period Pohlmann's alternative had better fuel consumption values.

7 | Conclusion

7.1 Summary

Even with the continuous advances in technology, enabling new approaches in traffic signal control, the current investigation opted to employ its efforts in obtaining better results, with the use of established technologies, for the short-term. The idea was to work on improvements for a current available ATCS that could be promptly implemented without restrictions. The chosen ATCS was the TUC strategy. The interest in TUC arose given its particular approach to traffic signal control. TUC employs a linear traffic flow model, which allows it to use an efficient algorithm that is capable of operating with high frequency update cycles and that, at the same time, is capable of considering the impact of its actions for the whole network, i.e. the network effect of every green time chosen is considered. This characteristic enables the control strategy to quickly react to localized changing traffic conditions disrupting the overall traffic operation the least possible.

Analyzing TUC's structure in more detail, a potential improvement point has been identified. TUC's approach to offset setting could be modified in order to allow it to better deal with different types of traffic networks, specifically meshed networks. Previously, TUC was only capable of maintaining offsets on routes whose green phases belonged to the beginning of the cycle. In meshed networks, there are many routes, crossing the main ones, that must be served by secondary green stages and that can also benefit from adjusted offsets. Moreover, TUC's method for implementing new offsets relies on the use of single transition signal plans that may incur in abrupt changes, which may negatively impact the operation of the network. The necessary transition plans are calculated regardless of the effect that they will have. A recent modification to the control technique employed by the Green Control Module (Aboudolas et al. 2009, 2010; de Oliveira and Camponogara 2007) offered the needed freedom demanded by the modifications proposed in the present work. The Model Predictive Control technique enables the use of constraints that limit the solution

space of the problem, and more important, these constraints may be modified for each optimization run of the algorithm. Aiming to solve both mentioned "weaknesses" of TUC, and taking advantage of the extra modeling capabilities offered by the MPC framework, the original traffic flow model was extended including offsets and cycles as additional variables. With this modification, it was not only possible to describe the secondary offsets, but it also transferred the task of determining the transition plans to the Green Control Module. Doing so, the impact of applying transition cycles will be weighted and the optimization will try to minimize this impact. Because of an unexpected behaviour of the new modeled system, the secondary offsets had to be indirectly codified, affecting the original proposition. Nevertheless, the problem was circumvented.

Aware of the fact that the modified traffic flow model used (extended from the original store-and-forward) is only capable of capturing the impact of changing cycles, and not offsets, a parallel path has been presented. Entitled Legacy, this second alternative achieves the same goals of first one, but with a much more straightforward approach. Instead of extending the traffic model to account for cycles and offsets, the Green Control Module was left unchanged. The changes were then directly applied to the Offset Control Module. In charge of imposing the transition cycles to the Green Control Module, the Offset Control Module was modified to successively apply transition cycles, not greater, nor shorter than 20% of the current common cycle length, until the desired offsets are reached. This guarantees smoother changes without actually testing their impact on the network.

Taking advantage of the open opportunity and driven by some interesting observations on the behaviour of queues during simulation runs, two unconventional methods for the determination of offsets were unpretentiously tested. Comparing the offsets being implemented by TUC and the offsets calculated with a technique for minimizing delays it was clear that both set of values were not similar. A series of tests in a simple two-way network with just two intersections showed that the simple maintenance of green-waves in the most demanded traffic route do not reflect in overall smaller average delays. Additionally, varying the traffic demands of the competing traffic streams showed that the offset value that would guarantee lower delays was also shifting towards the offset value that would have been chosen for the maintenance of green-waves in the most demanding traffic stream. Without further proof, two similar methods that combine the competing offsets according to the relative traffic demands were implemented.

The TUC strategy requires for its operation basically three types of information that dynamically change with time: queue lengths; turning-rates; and traffic demands. According to Papageorgiou 1995, TUC can handle variations in turning-rates without much disturbances in its performance. This means that the use of average values during its operation should not incur in erratic behaviour. The traffic demands may

be directly collected from the vehicle detectors as long as there is no ongoing congestion. Queue length, in the other hand, is the type of input data that can mostly disturb TUC's performance. Therefore, the current work also proposed a new method for estimating and predicting future queue lengths for the operation of TUC. The proposed method combines an Unscented Kalman Filter with a Cell Transmission Model in order to take advantage of the influence of the traffic network dynamics in the formation of the queues, and hence improve the queue estimations.

The last part of the current investigation consisted on the evaluation of all the developments achieved. Using a real traffic network, with traffic demands similar to the ones observed in real life, all the components were tested simulating a whole day of operation. At the end, a comparison between the modified TUC and an ATCS developed recently (POHLMANN 2010) was carried out.

7.2 Discussion

Starting with the Queue Predictor developed, it is clear that each step performed by the method was capable of gradually improving the results. Unfortunately, the final improvement measured in terms of the reduction of the average delay stayed at around 4–5%, depending on the local estimation method used. The use of real queue values, after being processed with noise, have demonstrated that the weakness lie probably on the average turning-rates being used. As the RMSE values show, the errors greater than 5 vehicles strike around 30% of the estimated/predicted values. Perhaps, the queue estimator would benefit from a turning-rate estimation step before its employment. Nevertheless, one must also consider the fact that the stochastic behaviour of traffic has a great influence in the results because of the frequency of the process. In order to better assess this problem, it would have been interesting to have investigated the variation in turning-rates between each cycle run in order to attest it.

Focusing on the modifications to TUC's offset setting strategy the results were also quite disappointing. The improvements brought by the modifications were really subtle lying around 2% in terms of average delay for normal traffic conditions, and around 4.5% for more demanding traffic conditions. The envisioned extensions, which allowed the transition cycles to be taken in consideration along with the calculation of the green times have not met the expectations. The results accomplished with the use of the other, much simpler, alternative were very similar. The small differences in relation to the original TUC implementation may be partially explained given the inherent characteristics of the traffic network used. A great percentage of the links of the network extend over 200 meters in length, which contributes to the absorption of the impacts caused by the abrupt changes in cycle lengths. The preliminary tests,

documented in Dantas and Friedrich 2012, which used a traffic network with links with much shorter links (around 100 meters in length), showed stronger improvements of around 10–20% depending on the traffic conditions. The ability to maintain secondary offsets had also a timid impact. This may be due the fact that the routes affected have relatively much lower traffic flowing. This may lead to two scenarios: either these routes are left out of the offset maintenance because they are part of a loop, and the weakest link gets discarded; or the granting of offset maintenance and the improvement in delays have little weight in the final results. This investigation would have been more interesting if a denser traffic network, with more intersections, have been used for the evaluation. But, the use of the present traffic network was interesting in the sense that it was possible to directly compare TUC's performance with the recently developed ATCS proposed in Pohlmann 2010.

The final comparison between the modified TUC, Pohlmann's ATCS and the fixed time control alternative has brought some interesting conclusions. The most visible is the fact that the fixed time alternative, optimized for the peak hours, had an almost unrivaled performance. With just two different sets of signal plans it was capable of keeping average delay levels even lower than TUC's. TUC's worse performance highlights the limits of its straightforward approach, where its linear traffic flow model seems to be the culprit. Nevertheless, it is still impressive what such strategy can accomplish. This comparison raises once more the question about the importance of automated and adaptive traffic signal control systems: Are they really necessary? The answer depends on the level of commitment of the public power in charge of the traffic network's maintenance. For the cases where the responsible department has a reserved budget for keeping track of the traffic evolution and performs updates to the signal plans in short intervals, e.g. every 5 years, than fixed time control seems to be the most adequate option, given its cheaper price and reliability. In these cases, the investment in monitoring infrastructure is enough. In the other hand, for different realities, specially the ones found in development countries, the budget is not always available and it may be more interesting to make a greater investment once and let the ATCS operate autonomously for a longer period of time.

A | Appendix

A.1 LQR

A discrete Linear Quadratic Regulator assumes a system represented by a set of linear difference equations:

$$x_{k+1} = \mathrm{A}x_k + \mathrm{B}u_k \tag{A.1}$$

with x being the state-variable vector; u the control-variable vector; and A and B mapping the dynamics. The control objective is achieved by the minimization of a quadratic function that weights both the primary objective and the effort of the necessary control actions:

$$J = \sum_{k=0}^{\infty} x_k^{\intercal}\mathrm{Q}x_k + u_k^{\intercal}\mathrm{R}u_k \tag{A.2}$$

The control sequence that minimizes the above cost function can be written as a feedback function:

$$u_k = -\mathrm{F}x_k \tag{A.3}$$

where:

$$\mathrm{F} = \left(\mathrm{R} + \mathrm{B}^{\intercal}\mathrm{P}\mathrm{B}\right)^{-1}\mathrm{B}^{\intercal}\mathrm{P}\mathrm{A} \tag{A.4}$$

and P, a positive definite matrix, is the solution of the Ricatti Difference Equation:

$$\mathrm{P} = \mathrm{Q} + \mathrm{A}^{\intercal}\left(\mathrm{P} - \mathrm{P}\mathrm{B}\left(\mathrm{R} + \mathrm{B}^{\intercal}\mathrm{P}\mathrm{B}\right)^{-1}\mathrm{B}^{\intercal}\mathrm{P}\right)\mathrm{A} \tag{A.5}$$

which can be numerically solved by successive iteration until P converges (Kwakernaak 1972). P must be calculated only once, and the control actions may be later on updated according only to the changing x_k values as shown in Equation A.3.

A.2 Cell Transmission Model

After the brief introduction given in Chapter 5, Section 5.2 Cell Transmission Model, it is also interesting to present two other features of the CTM: Merges and Diverges, which complement its model and allow the characterization of a real network, where different links may fuse into only one, and a given link may subdivide into others. For the sake of simplicity only the merge of two links into one, $2:1$, and the diverge of one link into two, $1:2$, as originally proposed in Daganzo 1995, will be presented. Pohlmann 2010 may be consulted for further $1:3$ and $3:1$ connections.

Recalling Equation 5.16:

$$q_i(k) = \mathbf{min}\left\{ n_i(k), q_i^{\max}, \frac{|v^{\mathrm{bw}}|}{v^{\mathrm{free}}}\left(n_{i+1}^{\max} - n_{i+1}(k)\right) \right\}$$

which rules the outflow of vehicles in each cell i, it is now interesting to divide it in three others:

$$S_i(k) = \mathbf{min}\left\{ n_i(k), q_i^{\max} \right\} \tag{A.6}$$

$$R_i(k) = \mathbf{min}\left\{ q_i^{\max}, \frac{|v^{\mathrm{bw}}|}{v^{\mathrm{free}}}\left(n_{i+1}^{\max} - n_{i+1}(k)\right) \right\} \tag{A.7}$$

$$q_i(k) = \mathbf{min}\left\{ S_i(k), R_{i+1}(k) \right\} \tag{A.8}$$

where:

$S_i(k)$ is the highest outflow that cell i can sustain during interval k, [veh]

$R_i(k)$ is the highest inflow that cell i can receive during interval k, [veh]

This separation is necessary for the description of the complementary equations for Merges and Diverges.

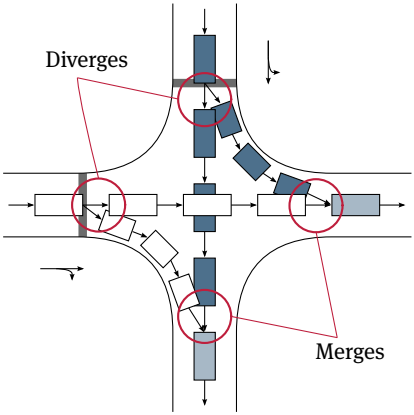


Figure A.1: CTM Merges and Diverges

A.2.1 Merges

A traffic network is made of the interconnection between multiple road stretches and these connection points must be contemplated by the traffic model. They are particularly important in intersections where it is necessary to merge the outflows of the traffic coming from the upstream links, as shown in the fictitious intersection example depicted in Figure A.1.

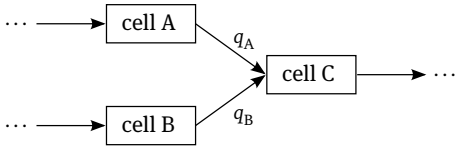


Figure A.2: Cell merge

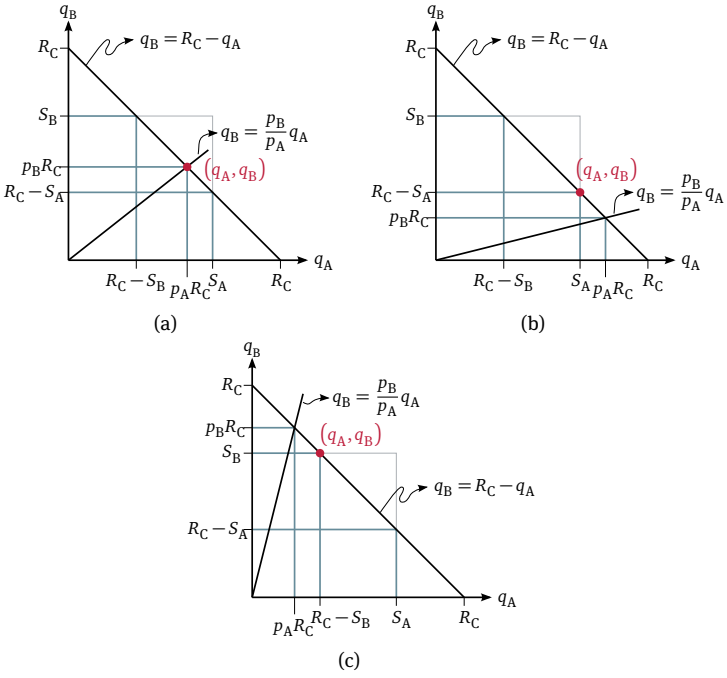


Figure A.3: Limited-inflow merge cases: $S_A + S_B > R_C$, (source: POHLMANN 2010)

Given a particular merge node, as shown in Figure A.2, it is necessary to handle how the two traffic flows, q_A and q_B, will be combined to enter cell C. The amount of vehicles that will be transferred from cells A and B during interval k cannot exceed the amount of vehicles that cell C can receive. Therefore, relying on Equations A.6, A.7 and A.8: $q_A + q_B \leqslant R_C$. Whenever the receiving capability of cell C is smaller than the total amount of vehicles that are coming from cells A and B: $R_C < S_A + S_B$, a priority scheme must be followed. According to each case, it is possible to allocate percentage priorities to each traffic flow, p_A and p_B with $p_A + p_B = 1.0$, that reflect the actual observed behaviour. Depending on the defined priorities three scenarios emerge, as shown in Figure A.3, and the solution will lie on the line described by $q_B = R_C - q_A$. Since the traffic flows q_A and q_B cannot be bigger than their respective sending potentials, S_A and S_B, the final solution will either lie on the limits, Figure A.3b and A.3c, or in between, Figure A.3a. This procedure of finding the allowed q_A and q_B for each interval k is summarized in Algorithm A.1 2 : 1 Cell Merge.

Algorithm A.1: 2 : 1 Cell Merge

mid$\{x, y, z\}$: function that receives three arguments, eliminates the highest and the lowest values and returns the remaining one;
if $S_A(k) + S_B(k) \leqslant R_C(k)$ **then**
    $q_A(k) = S_A(k)$;
    $q_B(k) = S_B(k)$;
else
    $q_A(k) = \textbf{mid}\{S_A(k), p_A R_C(k), R_C(k) - S_B(k)\}$;
    $q_B(k) = \textbf{mid}\{S_B(k), p_B R_C(k), R_C(k) - S_A(k)\}$;

A.2.2 Diverges

A Diverge node is much simpler than the Merge node. Looking at Figure A.4, the traffic flows coming from cell A, q_{A_1} and q_{A_2}, are easily defined according to the real turning percentages expected for the modeled diverge. Attaining to the principle that the outflow cannot exceed receiving capacity of the following cell, Equation A.8, the partial traffic flows are calculated accordingly:

$$q_A = \textbf{min}\left\{S_A, \frac{R_B}{t^{\vec{\uparrow}}_{A,B}}, \frac{R_C}{t^{\vec{\uparrow}}_{A,C}}\right\} \tag{A.9}$$

$$q_{A_1} = t^{\vec{\uparrow}}_{A,B} q_A \tag{A.10}$$

$$q_{A_2} = t^{\vec{\uparrow}}_{A,C} q_A \tag{A.11}$$

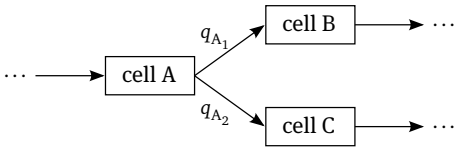


Figure A.4: Cell diverge

A.3 Standard Kalman Filter

The KF algorithm assumes a linear system model subject to noise:

$$\begin{aligned} X_{k+1} &= \boldsymbol{A} X_k + \boldsymbol{B} u_k + v_k \\ Y_k &= \boldsymbol{D}_k X_k + w_k \end{aligned} \tag{A.12}$$

where v is the process noise vector; and w is the measurement noise vector. This state-space representation is exactly the same one used throughout this work, except for the additional noise included. Each element of the noise vectors is a Gaussian random variable:

$$v_i \approx \mathcal{N}\left(0, \sigma_{v_i}^2\right) \tag{A.13}$$

$$w_j \approx \mathcal{N}\left(0, \sigma_{w_j}^2\right) \tag{A.14}$$

with $\mathcal{N}$ being a normal distribution with zero mean, $\mu = 0.0$, and variance σ^2, and since there is one noise variable associated to each state variable, the whole system can be considered a Gaussian random process. The current estimated values $\hat{X}$ are the expected values of the state variables X:

$$\hat{X}(k) = \mathbb{E}\big(X(k)\big) = \hat{X}_k \tag{A.15}$$

and the error between them is captured by the covariance matrix $P_{\boldsymbol{xx}}$:

$$P_{\boldsymbol{xx},k} = \mathbb{E}\big([X_k - \hat{X}_k][X_k - \hat{X}_k]^\intercal\big) \tag{A.16}$$

The relationship between each noise element v_i may also be depicted by a covariance matrix $\boldsymbol{Q}^{\boldsymbol{v}}$, which is actually diagonal, since they are supposed to be independently random. The same is valid for each w_i with the corresponding $\boldsymbol{Q}^{\boldsymbol{w}}$ covariance matrix:

$$\boldsymbol{Q}^{\boldsymbol{v}} = \begin{bmatrix} \sigma_{v_1}^2 & 0 & \cdots & 0 \\ 0 & \sigma_{v_2}^2 & & 0 \\ \vdots & & \ddots & \\ 0 & 0 & \cdots & \sigma_{v_i}^2 \end{bmatrix} \qquad \boldsymbol{Q}^{\boldsymbol{w}} = \begin{bmatrix} \sigma_{w_1}^2 & 0 & \cdots & 0 \\ 0 & \sigma_{w_2}^2 & & 0 \\ \vdots & & \ddots & \\ 0 & 0 & \cdots & \sigma_{w_j}^2 \end{bmatrix}$$

Depending on the case, matrix $\boldsymbol{Q}^{\boldsymbol{w}}$ may change size according to the number of measured variables, which can change during the operation of the system, i.e., $\boldsymbol{D}_k$ and $\boldsymbol{Q}_k^{\boldsymbol{w}}$.

Once the system has been parametrized, by defining its structure given in Equation A.12: $\boldsymbol{A}$; $\boldsymbol{B}$; $\boldsymbol{D}_k$; $\boldsymbol{Q}^{\boldsymbol{v}}$; $\boldsymbol{Q}_k^{\boldsymbol{w}}$, and expected initial values $\hat{X}_0$ and $P_{\boldsymbol{xx},0}$, the Algorithm A.2 Standard Kalman Filter may be called at each new sample time to estimate the current state of the system. The initial values for $\hat{X}_0$ and $P_{\boldsymbol{xx},0}$, along with $\boldsymbol{Q}^{\boldsymbol{v}}$ and $\boldsymbol{Q}^{\boldsymbol{w}}$ may be guessed and tuned through trial and error until the expected operation quality is achieved. As explained in Chapter 5 Queue Estimation, Section 5.3 Unscented Kalman Filter, the algorithm propagates in time the state of the system, based on its last estimation, and, combining it with the current available measured states, it is capable of offering a current system estimate with the least expected error (Anderson

and MOORE 2012).

Algorithm A.2: Standard Kalman Filter

$\hat{X}$: estimated values of the state variables;
$P_{\boldsymbol{xx}}$: predicted estimate covariance matrix;
Y : measured values of the state variables;
$\tilde{Y}$: measurement residual;
$P_{\boldsymbol{yy}}$: residual covariance matrix;
$\boldsymbol{K}$: kalman gain;
Time Update:

$$\hat{X}_{k|k-1} \leftarrow \boldsymbol{A}\hat{X}_{k-1} + \boldsymbol{B}u_{k-1};$$
$$P_{\boldsymbol{xx},k|k-1} \leftarrow \boldsymbol{A}P_{\boldsymbol{xx},k-1}\boldsymbol{A}^{\intercal} + \boldsymbol{Q}^{\boldsymbol{v}};$$

Measurement Update:

$$\tilde{Y}_k \leftarrow Y_k - \boldsymbol{D}_k\hat{X}_{k|k-1};$$
$$P_{\boldsymbol{yy},k} \leftarrow \boldsymbol{D}_k P_{\boldsymbol{xx},k|k-1}\boldsymbol{D}_k^{\intercal} + \boldsymbol{Q}_k^{\boldsymbol{w}};$$
$$\boldsymbol{K}_k \leftarrow P_{\boldsymbol{xx},k|k-1}\boldsymbol{D}_k^{\intercal}P_{\boldsymbol{yy},k}^{-1};$$
$$\hat{X}_k \leftarrow \hat{X}_{k|k-1} + \boldsymbol{K}_k\tilde{Y}_k;$$
$$P_{\boldsymbol{xx},k} \leftarrow \left(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{D}_k\right)P_{\boldsymbol{xx},k|k-1};$$

A.4 Unscented Kalman Filter

The UKF has been developed with nonlinear systems in mind with which the standard KF cannot be applied. In nonlinear systems the state space representation changes to:

$$\begin{aligned} X_{k+1} &= \boldsymbol{f}(X_k, u_k, v_k) \\ Y_k &= \boldsymbol{h}(X_k, w_k) \end{aligned} \tag{A.17}$$

where $\boldsymbol{f}$ is the nonlinear function that maps the system dynamics, and $\boldsymbol{h}$ maps the measured states. The UKF is able to capture the future probability distribution of the system by evaluating a predefined number of possible current points of operation, the *sigma-points*, that together represent the current expected mean values, related to each state variable, and covariance matrix of the system. Once these operation points are propagated through the system's nonlinear model, the future resulting points represent, thus, the future probability distribution of the system.

As explained in WAN and VAN DER MERWE 2001, the system must be augmented according to the size of each noise vector, v_k and w_k, which transforms the state vector

$\hat{X}$ and covariance matrix P into:

$$\hat{X}^{\mathbf{a}}(k) = \mathbb{E}\big(X^{\mathbf{a}}(k)\big) = \begin{bmatrix} \hat{X}_k^{\intercal} & \mathbf{0}_{1\times n^{\boldsymbol{x}}} & \mathbf{0}_{1\times n_k^{\boldsymbol{w}}} \end{bmatrix}^{\intercal} = \hat{X}_k^{\mathbf{a}} \tag{A.18}$$

$$P^{\mathbf{a}}(k) = \mathbb{E}\big([X_k^{\mathbf{a}} - \hat{X}_k^{\mathbf{a}}][X_k^{\mathbf{a}} - \hat{X}_k^{\mathbf{a}}]^{\intercal}\big) = \begin{bmatrix} P_{\boldsymbol{xx},k} & 0 & 0 \\ 0 & \boldsymbol{Q}^{\boldsymbol{v}} & 0 \\ 0 & 0 & \boldsymbol{Q}_k^{\boldsymbol{w}} \end{bmatrix} = P_k^{\mathbf{a}} \tag{A.19}$$

where $\boldsymbol{Q}^{\boldsymbol{v}}$ and $\boldsymbol{Q}_k^{\boldsymbol{w}}$ are the covariance matrices of v_k and w_k respectively. The values of $\hat{X}_0$, $P_{\boldsymbol{xx},0}$, $\boldsymbol{Q}^{\boldsymbol{v}}$ and $\boldsymbol{Q}_0^{\boldsymbol{w}}$ are guessed initially and may be tuned later for better results.

Then, the *sigma-points* are generated according to:

$$\chi_{0,k} = \hat{X}_k^{\mathbf{a}} \tag{A.20}$$

$$\chi_{i,k} = \hat{X}_k^{\mathbf{a}} + \left(\gamma\sqrt{P_k^{\mathbf{a}}}\right)_i \qquad i = 1, \ldots, L^{\mathbf{a}} \tag{A.21}$$

$$\chi_{i,k} = \hat{X}_k^{\mathbf{a}} - \left(\gamma\sqrt{P_k^{\mathbf{a}}}\right)_{i-L^{\mathbf{a}}} \quad i = L^{\mathbf{a}} + 1, \ldots, 2L^{\mathbf{a}} \tag{A.22}$$

where $\gamma = \sqrt{L^{\mathbf{a}} + \lambda}$; $L^{\mathbf{a}}$ is size of the augmented system; $\lambda = \alpha^2(L^{\mathbf{a}} + \kappa) - L^{\mathbf{a}}$; and $\left(\gamma\sqrt{P_k^{\mathbf{a}}}\right)_i$ is the i^{th} column of $\gamma\sqrt{P_k^{\mathbf{a}}}$. The constants α and κ regulate the spread of the *sigma-points* around $\hat{X}$, where α is usually set to $10^{-4} \leqslant \alpha \leqslant 1$, and κ to 0 or $3 - L^{\mathbf{a}}$ (VAN DER MERWE 2004). The resulting *sigma-points* matrix:

$$\boldsymbol{\chi}_{k-1} = \begin{bmatrix} \hat{X}_{k-1}^{\mathbf{a}} & \hat{X}_{k-1}^{\mathbf{a}} + \gamma\sqrt{P_{k-1}^{\mathbf{a}}} & \hat{X}_{k-1}^{\mathbf{a}} - \gamma\sqrt{P_{k-1}^{\mathbf{a}}} \end{bmatrix} \tag{A.23}$$

may be also depicted as:

$$\boldsymbol{\chi} = \begin{bmatrix} [\boldsymbol{\chi}^{\boldsymbol{x}}]^{\intercal} & [\boldsymbol{\chi}^{\boldsymbol{v}}]^{\intercal} & [\boldsymbol{\chi}^{\boldsymbol{w}}]^{\intercal} \end{bmatrix}^{\intercal} \tag{A.24}$$

in order to highlight each of its components that will be used ahead. The *sigma-points* are evaluated with a weighting scheme that captures the current mean and covariance of the system:

$$\boldsymbol{\chi}_{k|k-1}^{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{\chi}_{k-1}^{\boldsymbol{x}}, u_{k-1}, \boldsymbol{\chi}_{k-1}^{\boldsymbol{v}}) \tag{A.25}$$

$$\hat{X}_k^{-} = \sum_{i=0}^{2L^{\mathbf{a}}} W_i^{\mathbf{m}} \boldsymbol{\chi}_{i,k|k-1}^{\boldsymbol{x}} \tag{A.26}$$

$$P^{-}_{\boldsymbol{xx},k} = \sum_{i=0}^{2L^{\mathbf{a}}} W_i^{\mathbf{c}} \left[\chi^{\boldsymbol{x}}_{i,k|k-1} - \hat{X}^{-}_{k}\right]\left[\chi^{\boldsymbol{x}}_{i,k|k-1} - \hat{X}^{-}_{k}\right]^{\intercal} \tag{A.27}$$

where $\chi^{\boldsymbol{x}}_{i,k|k-1}$ is the i^{th} column of $\boldsymbol{\chi}^{\boldsymbol{x}}_{k|k-1}$ and:

$$W_0^{\mathbf{m}} = \lambda/(L^{\mathbf{a}} + \lambda) \tag{A.28}$$

$$W_0^{\mathbf{c}} = \lambda/(L^{\mathbf{a}} + \lambda) + (1 + \alpha^2 + \beta) \tag{A.29}$$

$$W_i^{\mathbf{m}} = W_i^{\mathbf{c}} = 1\Big/\big(2(L^{\mathbf{a}} + \lambda)\big) \quad i = 1, \ldots, 2L^{\mathbf{a}} \tag{A.30}$$

where the constant β is usually set to 2, and is used to account for the prior knowledge of the distribution of X. This weighting scheme is also used in the posterior update for capturing the expected future mean and covariances of the system:

$$\boldsymbol{\Upsilon}_{k|k-1} = \boldsymbol{h}(\boldsymbol{\chi}^{\boldsymbol{x}}_{k|k-1}, \boldsymbol{\chi}^{\boldsymbol{w}}_{k|k-1}) \tag{A.31}$$

$$\hat{Y}^{-}_{k} = \sum_{i=0}^{2L^{\mathbf{a}}} W_i^{\mathbf{m}} \Upsilon_{i,k|k-1} \tag{A.32}$$

$$P_{\boldsymbol{yy},k} = \sum_{i=0}^{2L^{\mathbf{a}}} W_i^{\mathbf{c}} \left[\Upsilon_{i,k|k-1} - \hat{Y}^{-}_{k}\right]\left[\Upsilon_{i,k|k-1} - \hat{Y}^{-}_{k}\right]^{\intercal} \tag{A.33}$$

$$P_{\boldsymbol{xy},k} = \sum_{i=0}^{2L^{\mathbf{a}}} W_i^{\mathbf{c}} \left[\chi^{\boldsymbol{x}}_{i,k|k-1} - \hat{X}^{-}_{k}\right]\left[\Upsilon_{i,k|k-1} - \hat{Y}^{-}_{k}\right]^{\intercal} \tag{A.34}$$

which are finally combined with the available real measurements Y_k:

$$\boldsymbol{K}_k = P_{\boldsymbol{xy},k}\left[P_{\boldsymbol{yy},k}\right]^{-1} \tag{A.35}$$

$$\hat{X}_k = \hat{X}^{-}_{k} + \boldsymbol{K}_k\left[Y_k - \hat{Y}^{-}_{k}\right] \tag{A.36}$$

$$P_{\boldsymbol{xx},k} = P^{-}_{\boldsymbol{xx},k} - \boldsymbol{K}_k P_{\boldsymbol{yy},k}\left[\boldsymbol{K}_k\right]^{\intercal} \tag{A.37}$$

and generate the filtered values of $\hat{X}_k$ and the updated covariance matrix $P_{\boldsymbol{xx},k}$. The

whole process takes place at each new sample time, and is summarized in Algorithm A.3 Unscented Kalman Filter.

Algorithm A.3: Unscented Kalman Filter

$\hat{X}$: estimated values of the state variables;
$P_{\boldsymbol{xx}}$: predicted estimate covariance matrix;
$\boldsymbol{\chi}$: sigma matrix;
Y : measured values of the state variables;
$\tilde{Y}$: measurement residual;
$P_{\boldsymbol{yy}}$: residual covariance matrix;
$P_{\boldsymbol{xy}}$: cross-covariance matrix;
$\boldsymbol{K}$: kalman gain;

Augment States:

$$\hat{X}^{\mathbf{a}}_{k-1} \leftarrow \begin{bmatrix} \hat{X}^{\intercal}_{k-1} & \mathbf{0}_{1\times n^x} & \mathbf{0}_{1\times n^w_k} \end{bmatrix}^{\intercal};$$

$$P^{\mathbf{a}}_{k-1} \leftarrow \begin{bmatrix} P_{\boldsymbol{xx},k-1} & 0 & 0 \\ 0 & \boldsymbol{Q}^{\boldsymbol{v}} & 0 \\ 0 & 0 & \boldsymbol{Q}^{\boldsymbol{w}}_k \end{bmatrix};$$

Generate *Sigma-Points:*

$$\boldsymbol{\chi}_{k-1} \leftarrow \begin{bmatrix} \hat{X}^{\mathbf{a}}_{k-1} & \hat{X}^{\mathbf{a}}_{k-1} + \gamma\sqrt{P^{\mathbf{a}}_{k-1}} & \hat{X}^{\mathbf{a}}_{k-1} - \gamma\sqrt{P^{\mathbf{a}}_{k-1}} \end{bmatrix}$$

Time Update:

$$\boldsymbol{\chi}_{k|k-1} \leftarrow \boldsymbol{f}(\boldsymbol{\chi}^{\boldsymbol{x}}_{k-1}, u_{k-1}, \boldsymbol{\chi}^{\boldsymbol{v}}_{k-1});$$

$$\hat{X}^{-}_{k} \leftarrow \sum_{i=0}^{2L^{\mathbf{a}}} W^{\mathbf{m}}_i \boldsymbol{\chi}^{\boldsymbol{x}}_{i,k|k-1};$$

$$P^{-}_{\boldsymbol{xx},k} \leftarrow \sum_{i=0}^{2L^{\mathbf{a}}} W^{\mathbf{c}}_i \left[\boldsymbol{\chi}^{\boldsymbol{x}}_{i,k|k-1} - \hat{X}^{-}_{k}\right]\left[\boldsymbol{\chi}^{\boldsymbol{x}}_{i,k|k-1} - \hat{X}^{-}_{k}\right]^{\intercal};$$

$$\boldsymbol{\Upsilon}_{k|k-1} \leftarrow \boldsymbol{h}(\boldsymbol{\chi}^{\boldsymbol{x}}_{k|k-1}, \boldsymbol{\chi}^{\boldsymbol{w}}_{k|k-1});$$

$$\hat{Y}^{-}_{k} \leftarrow \sum_{i=0}^{2L^{\mathbf{a}}} W^{\mathbf{m}}_i \Upsilon_{i,k|k-1};$$

Measurement Update:

$$P_{\boldsymbol{yy},k} \leftarrow \sum_{i=0}^{2L^{\mathbf{a}}} W^{\mathbf{c}}_i \left[\Upsilon_{i,k|k-1} - \hat{Y}^{-}_{k}\right]\left[\Upsilon_{i,k|k-1} - \hat{Y}^{-}_{k}\right]^{\intercal};$$

$$P_{\boldsymbol{xy},k} \leftarrow \sum_{i=0}^{2L^{\mathbf{a}}} W^{\mathbf{c}}_i \left[\boldsymbol{\chi}^{\boldsymbol{x}}_{i,k|k-1} - \hat{X}^{-}_{k}\right]\left[\Upsilon_{i,k|k-1} - \hat{Y}^{-}_{k}\right]^{\intercal};$$

$$\boldsymbol{K}_k \leftarrow P_{\boldsymbol{xy},k}\left[P_{\boldsymbol{yy},k}\right]^{-1};$$

$$\hat{X}_k \leftarrow \hat{X}^{-}_{k} + \boldsymbol{K}_k\left[Y_k - \hat{Y}^{-}_{k}\right];$$

$$P_{\boldsymbol{xx},k} \leftarrow P^{-}_{\boldsymbol{xx},k} - \boldsymbol{K}_k P_{\boldsymbol{yy},k}\left[\boldsymbol{K}_k\right]^{\intercal};$$

A.5 Performance Comparison of the Statistical Indicators Used

In order to better interpret the results of the evaluation of the Queue Predictor proposed in this work, a simple comparison test of the statistical indicators used has been carried out. As shown in Figure A.5, three different sets of nine values (A, B and C) are plotted against a Reference. Set A evolves like the Reference, attaining the same sign of the Reference's slopes, but usually "over-estimates" the values by 5 units. Set B follows the mean tendency of the Reference, and also presents an error of 5 units in most cases. Finally, set C has exactly the same slopes of the Reference, but has a constant offset of 5 units.

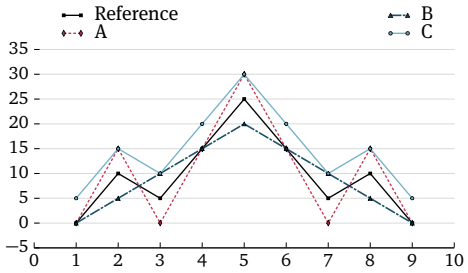


Figure A.5: Hypothetical reference following cases

Summarized in Table A.1 are the resulting RMSE, ReMSE and r values. Sets A and B hold the same and best values for RMSE and ReMSE, even though their behaviour are quite different. Set C, as expected, has a correlation coefficient of 1.0. The conclusion is that these indicators may hide what is really happening, and that, for the case of the Queue Predictor being evaluated, the actual performance of system must be examined to draw the final conclusions. These indicators help to make the decision when choosing the best alternative, but alone they may not lead to the right choice.

Alternative	RMSE [?]	ReMSE [-]	r [-]
A	3.7268	0.3946	0.9481
B	3.7268	0.3946	0.8753
C	5.0	0.5294	1.0

Table A.1: Comparison of the statistical indicators

A.6 Tests with Pohlmann's ATCS

Even though the focus of the present work is the TUC Strategy, the fact that the ATCS presented in Pohlmann 2010 has been used as reference case opens a window for analysing its performance. The tests that were carried out using Pohlmann's solution were also important for understanding some aspects of TUC and helped to draw the final conclusions of the present work. Apart from that, this Appendix remains as a complementary documentation, and a revision, of Pohlmann's ATCS performance.

Because the current work used a different version of the simulation software, AIMSUN, and also a modified version of the simulation scenario itself, part of the simulation results of Pohlmann's work had to be recalculated through new simulation runs.

The first batch of tests were related to the operation of the system depending on the cycle control method used (see Section 3.2, Chapter 3). During this step, an error was found in the code. Mistakingly, the green times were getting 2 additional seconds, which were being subtracted from the interstages, therefore keeping the cycle length unchanged. This error had a great impact in the final results regarding the use of either of the two methods proposed, the Webster's formula based cycles and the Saturation Based cycles. Comparing the cycle lengths depicted in Figure A.6 and the original ones in Pohlmann 2010, there is not much difference, but as Figure A.7 shows, the system has a completely different behaviour. The system actually has a much better performance using Webster cycles. The smaller cycles produced by the Saturation Based method caused real problems during the peak hours, which also contaminated the performance during the light traffic period, given the extended recovery period.

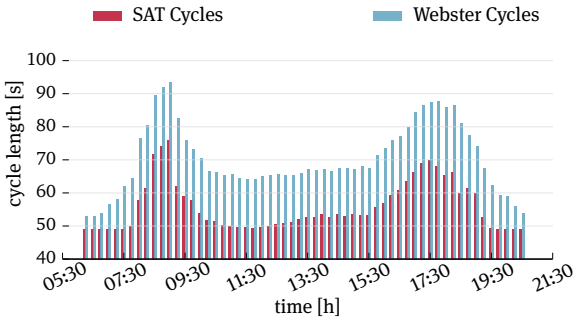


Figure A.6: Webster and SAT reference cycles

Another question raised by working with TUC was about the actual advantage of its frequent modification of traffic signal plans. It is clear that this feature contributes

to a more responsive traffic control system, capable of reacting faster to changes in traffic conditions. But, given the better results offered by Pohlmann's solution, there was also an interest in finding out why exactly this happened.

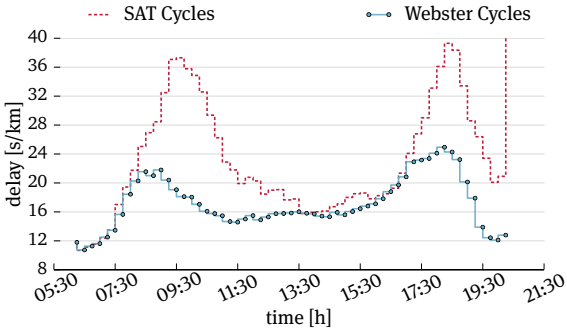


Figure A.7: Webster vs. SAT delays

With the control update frequency in mind, Pohlmann's strategy was modified to operate in two different frequencies in order to analyse the results. As mentioned in Section 2.1.6, Chapter 2, the chosen update period was set to 15 minutes, but there are not any tests mentioned that might have been carried out in order to validate this 15-minute value as the best option. Given the expectation that a more frequent update would possibly improve the performance, and a less frequent update would degrade the performance, tests with 7.5-minute and 30-minute update periods were conducted.

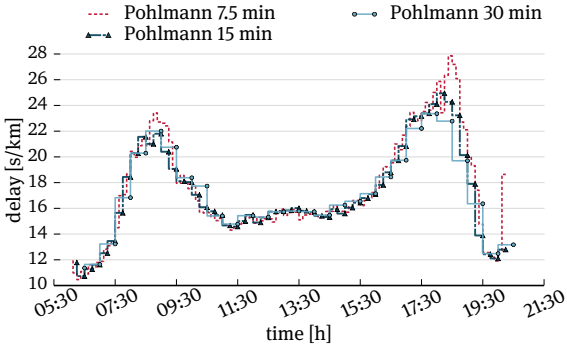


Figure A.8: Update periods multiple of 7.5 minutes

Opposed to the expectation, the system performed worse with the shorter update pe-

riod and slightly better with the longer one as depicted in Figure A.8. The change in the length of the update period also implies in a different length of the aggregated traffic demand data stored in the database. This means that, leaving the rest of the technique unchanged, the algorithm was forced to match a much shorter sample of the traffic demand stored in the database than before, for the case of the 7.5-minute update period (see Section 2.1.6 in Chapter 2 Literature Review). This could explain part of the reason why the shorter update period resulted in a worse performance, but the error rate in the matching procedure did not increase. The culprit seems to be the time period during which the algorithm evaluates each offset possibility, which is also equal to the update period. Since this process also includes the transition cycles necessary to achieve the offsets being tested, the final offsets may have not been given enough time to be effectively tested. This may have led the algorithm to choose offsets that incurred in less delay during the transition cycles but not necessarily in better network performance given the offsets alone. Either way, the performance remained unchanged during the light traffic periods for all the options tested.

The above investigation also induced some questioning about the algorithm's sensibility in relation to the changing traffic demands throughout the day. The chosen update period is exactly the same as the size of the different O/D matrices employed, i.e. the length of each traffic demand period that compose the final traffic demand profile of the simulation scenario. This means that there is a synchronization between the change in traffic demands in the simulation scenario and the search for a matching traffic demand profile in the database. This coincidence might have also aided in conquering better results. Note that there is a difference between aggregating data in a given fixed time length and expecting that the traffic conditions will change in real life at evenly distributed time periods. In order to test this hypothesis, two additional update periods were tested: a 14-minute and a 16-minute update period. This guarantees that both changing traffic conditions and data aggregation will not be synchronized anymore. As Figure A.9 shows, there was actually almost no difference in terms of traffic delay between the alternatives. But, at the second peak period it is evident that the original configuration was beneficial to the system's performance and that this type of detail must also be taken into consideration when evaluating such systems.

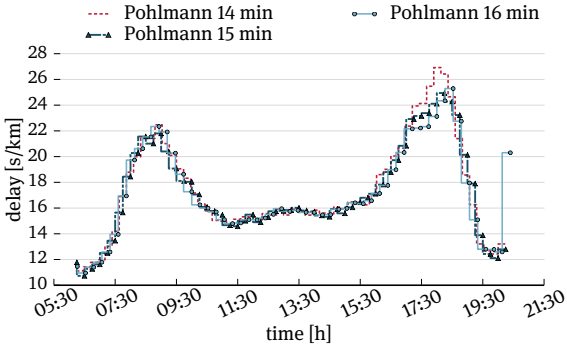


Figure A.9: Unsynchronized update periods

Bibliography

Aboudolas, K., M. Papageorgiou, and E. Kosmatopoulos (2009). "Store-and-Forward Based Methods for the Signal Control Problem in Large-scale Congested Urban Road Networks". In: *Transportation Research Part C: Emerging Technologies* 17.2, pp. 163–174. ISSN: 0968-090X (cited in pages 19, 28, 30, 93).

Aboudolas, K., M. Papageorgiou, A. Kouvelas, and E. Kosmatopoulos (2010). "A Rolling-horizon Quadratic-programming Approach to the Signal Control Problem in Large-scale Congested Urban Road Networks". In: *Transportation Research Part C: Emerging Technologies* 18.5, pp. 680–694. ISSN: 0968-090X (cited in pages 19, 30, 93).

Abu-Lebdeh, G. and R. Benekohal (1997). "Development of Traffic Control and Queue Management Procedures for Oversaturated Arterials". In: *Transportation Research Record* 1603.1, pp. 119–127. ISSN: 0361-1981 (cited in page 25).

Almasri, E. (2006). "A New Offset Optimization Method for Signalized Urban Road Networks". PhD thesis. Hanover, Germany: Leibniz Universität Hannover (cited in page 17).

Anderson, B. D. O. and J. B. Moore (2012). *Optimal Filtering*. Mineola, NY: Dover Publications, Incorporated. ISBN: 9780486136899 (cited in pages 58, 67, 102).

Ansorge, R. (1990). "What Does the Entropy Condition Mean in Traffic Flow Theory?" In: *Transportation Research Part B: Methodological* 24.2, pp. 133–143. ISSN: 0191-2615 (cited in page 62).

Bellman, R. (1954). "The Theory of Dynamic Programming". In: *Bulletin of the American Mathematical Society* 60.6, pp. 503–515. ISSN: 0002-9904, 1936-881X (cited in page 15).

Braun, R. and C. Kemper (2008). "GALOP-Online — Ein Genetischer Algorithmus zur netzweiten Online-Optimierung der Lichtsignalsteuerung [A Genetic Algorithm for Network-wide Online Optimization of Traffic Signals]". German. In: *HEUREKA*

'08 — Optimierung in Verkehr und Transport. 002/90. Stuttgart: FGSV — Forschungsgesellschaft für Straßen-und Verkehrswesen. ISBN: 978-3-939715-48-1 (cited in page 11).

Braun, R., C. Kemper, and F. Weichenmeier (2008). "TRAVOLUTION — Adaptive Urban Traffic Signal Control with an Evolutionary Algorithm". In: *Proceedings of the 4th International Symposium Networks for Mobility*. Stuttgart, Germany: FOVUS — Forschungsschwerpunkt Verkehr Universität Stuttgart (cited in page 11).

Camacho, E. F. and C. A. Bordons (2004). *Model Predictive Control*. 2nd. Springer. ISBN: 978-1-85233-694-3 (cited in pages 30, 31).

Catling, I. and R. Harris (1995). "TABASCO — Improving Transport Systems in Europe". In: *Proceedings of the 1995 Vehicle Navigation and Information Systems Conference — In conjunction with the Pacific Rim TransTech Conference. 6th International VNIS. 'A Ride into the Future'*, pp. 503–507 (cited in pages 11, 17).

Chen, C., J. Hu, and Y. Wang (2010). "Cell-based Simulation and Estimation of Urban Traffic Network". In: *13th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 7–12. ISBN: 978-1-4244-7657-2 (cited in page 58).

Csallner, A., H. G. Schlichter, and B. Ziegler (1995). "Entwicklung und Organisation des Projekts Munich COMFORT [The Development and Organization of the Munich Project COMFORT]". German. In: *Internationales Verkehrswesen* 47.9. ISSN: 0020-9511 (cited in page 11).

Daganzo, C. F. (1994). *The Cell Transmission Model: Network Traffic*. Working Papers. Institute of Transportation Studies, UC Berkeley (cited in pages 17, 62).

— (1995). "The Cell Transmission Model, Part II: Network Traffic". In: *Transportation Research Part B: Methodological* 29.2, pp. 79–93. ISSN: 0191-2615 (cited in pages 62, 98).

Dantas, L. D. and B. Friedrich (2012). "Concurrent Split, Offset and Cycle Control through Model Predictive Control". In: *Proceedings of the 13th IFAC Symposium on Control in Transportation Systems*. Sofia (cited in pages 85, 96).

— (2013). "Short-term Online Queue Predictor Based on Cell Transmission Model and Unscented Kalman Filter". In: *TRB 92nd Annual Meeting Compendium of Papers*. Washington DC, USA: Transportation Research Board (cited in pages 60, 69, 71).

de Oliveira, L. B. and E. Camponogara (2007). "Predictive Control for Urban Traffic Networks: Initial Evaluation". In: *Proceedings of the 3rd IFAC Symposium on System Structure and Control*. Iguassu Falls, Brazil (cited in pages 30, 93).

DELL'OLMO, P. and P. B. MIRCHANDANI (1995). "REALBAND: An Approach for Real-time Coordination of Traffic Flows on Networks". In: *Transportation Research Record* 1494. ISSN: 0361-1981 (cited in page 14).

— (1996). "A Model for Real-Time Traffic Coordination Using Simulation Based Optimization". In: *Advanced Methods in Transportation Analysis*. Transportation Analysis. Springer Berlin–Heidelberg, pp. 525–546. ISBN: 978-3-642-85258-9, 978-3-642-85256-5 (cited in page 14).

DIAKAKI, C. (1999). "Integrated Control Of Traffic Flow In Corridor Networks". PhD thesis. Chania, Greece: Technical University Of Crete (cited in pages 57, 58).

DIAKAKI, C., V. DINOPOULOU, K. ABOUDOLAS, M. PAPAGEORGIOU, E. BEN-SHABAT, E. SEIDER, and A. LEIBOV (2003). "Extensions and New Applications of the Traffic-responsive Urban Control Strategy: Coordinated Signal Control for Urban Networks". In: *Transportation Research Record: Journal of the Transportation Research Board* 1856.-1, pp. 202–211 (cited in pages 17, 19, 20, 25, 26).

DIAKAKI, C., M. PAPAGEORGIOU, and K. ABOUDOLAS (2002). "A Multivariable Regulator Approach to Traffic-responsive Network-wide Signal Control". In: *Control Engineering Practice* 10.2, pp. 183–195. ISSN: 0967-0661 (cited in pages 17, 19, 28).

FÖRSTER, G. (2008). "Kurzfristprognose auf Basis von Raum-Zeit-Mustern [Short-term Prediction Based on Space-time Patterns]". German. In: *HEUREKA '08 — Optimierung in Verkehr und Transport*. ⁰⁰²⁄₉₀. FGSV — Forschungsgesellschaft für Straßen-und Verkehrswesen. ISBN: 978-3-939715-48-1 (cited in page 16).

FRIEDRICH, B. (1999). "Ein verkehrsadaptives Verfahren zur Steuerung von Lichtsignalanlagen [A Traffic-adaptive Method for the Control of Traffic Signals]". German. PhD thesis. Munich, Germany: Technische Universität München (cited in page 11).

FRIEDRICH, B. and Y.-P. WANG (2006). "Optimizing Origin-Destination (O-D) Estimation with Respect to Redundant Information and Route Choice". In: *11th IFAC Symposium on Control in Transportation Systems*. Vol. 11. 1. Netherlands, pp. 633–637 (cited in page 16).

— (2008). "Optimierung der Matrixschätzung durch Elimination redundanter Informationen [Optimizing matrix estimation through the elimination of redundant information]". German. In: *HEUREKA '08 — Optimierung in Verkehr und Transport*. ⁰⁰²⁄₉₀. FGSV — Forschungsgesellschaft für Straßen-und Verkehrswesen. ISBN: 978-3-939715-48-1 (cited in page 16).

GANG, L., G. JIANG, and Z. CAI (2007). "Traffic State Estimation Method for Arterial Street". In: *International Conference on Transportation Engineering 2007, ICTE 2007*, pp. 443–448 (cited in pages 57, 58).

Gartner, N. H. (1982). “Development and Testing of a Demand-responsive Strategy for Traffic Signal Control”. In: *American Control Conference, 1982*. Arlington, VA, USA, pp. 578–583 (cited in page 12).

— (1983). “OPAC: A Demand-responsive Strategy for Traffic Signal Control”. In: *Transportation Research Record* 906, pp. 75–81. ISSN: 0361-1981 (cited in page 12).

— (1984). “Development of Demand-responsive Strategies for Urban Traffic Control”. In: *System Modelling and Optimization*. Lecture Notes in Control and Information Sciences 59. Springer Berlin–Heidelberg, pp. 166–174. ISBN: 978-3-540-13185-4, 978-3-540-38828-9 (cited in page 13).

Gartner, N. H., M. H. Kaltenbach, and M. M. Miyamoto (1983). *Demand-responsive Decentralized Urban Traffic Control Part II: Network Extensions*. Tech. rep. DOT/OST/P-34/85/009. Washington, D.C.: U.S. Dept. of Transportation (cited in page 12).

Gartner, N. H., F. Pooran, and C. Andrews (2001). “Implementation of the OPAC Adaptive Control Strategy in a Traffic Signal Network”. In: *2001 IEEE Intelligent Transportation Systems Conference Proceedings*, pp. 195–200 (cited in pages 12, 13).

Gazis, D. C. and R. B. Potts (1963). “The Oversaturated Intersection”. In: *Proceedings 2nd International Symposium on the Theory of Traffic Flow*. London, pp. 221–237 (cited in page 28).

Gordon, N., D. Salmond, and A. F. M. Smith (1993). “Novel Approach to Nonlinear / non-Gaussian Bayesian State Estimation”. In: *Radar and Signal Processing, IEE Proceedings F* 140.2, pp. 107–113. ISSN: 0956-375X (cited in page 68).

HBS (2009). *Handbuch für die Bemessung von Straßenverkehrsanlagen [Manual for the Dimensioning of Road Infrastructure]*. German. FGSV — Forschungsgesellschaft für Straßen-und Verkehrswesen. ISBN: 978-3-941790-35-3 (cited in pages 17, 20).

Head, K. L. (1995). “Event-based Short-term Traffic Flow Prediction Model”. In: *Transportation Research Record* 1510, pp. 45–52. ISSN: 0361-1981 (cited in page 15).

Head, K. L., P. B. Mirchandani, and D. Sheppard (1992). “Hierarchical Framework for Real-time Traffic Control”. In: *Transportation Research Record* 1360, pp. 82–88. ISSN: 0361-1981 (cited in page 13).

Hu, X., W. Wang, and H. Sheng (2010). “Urban Traffic Flow Prediction with Variable Cell Transmission Model”. In: *Journal of Transportation Systems Engineering and Information Technology* 10.4, pp. 73–78. ISSN: 1570-6672 (cited in page 58).

Hunt, P. B., D. I. Robertson, and R. D. Bretherton (1982). "The SCOOT On-line Traffic Signal Optimisation Technique". In: *Traffic Engineering and Control* 23, pp. 190–192 (cited in page 7).

Hunt, P. B., D. I. Robertson, R. D. Bretherton, and R. I. Winton (1981). *SCOOT — A Traffic Responsive Method of Coordinating Signals*. TRRL Laboratory Report 1014. Transport and Road Research Laboratory, pp. 1–41 (cited in page 7).

Jabari, S. E. and H. X. Liu (2012). "A Stochastic Model of Traffic Flow: Theoretical Foundations". In: *Transportation Research Part B: Methodological* 46.1, pp. 156–174. ISSN: 0191-2615 (cited in pages 58, 67).

— (2013). "A Stochastic Model of Traffic Flow: Gaussian Approximation and Estimation". In: *Transportation Research Part B: Methodological* 47, pp. 15–41. ISSN: 0191-2615 (cited in pages 57, 67).

Julier, S. J., J. K. Uhlmann, and H. F. Durrant-Whyte (1995). "A New Approach for Filtering Nonlinear Systems". In: *Proceedings of the 1995 American Control Conference*. Vol. 3. IEEE, pp. 1628–1632. ISBN: 0-7803-2445-5 (cited in pages 58, 68).

Julier, S. J. and J. K. Uhlmann (1997). "A New Extension of the Kalman Filter to Nonlinear Systems". In: *International Symposium on Aerospace/Defense Sensing, Simulation and Controls*. Orlando, FL, USA (cited in pages 58, 68).

Kailath, T. (1974). "A View of Three Decades of Linear Filtering Theory". In: *IEEE Transactions on Information Theory* 20.2, pp. 146–181. ISSN: 0018-9448 (cited in page 66).

Kalman, R. E. (1960). "A New Approach to Linear Filtering and Prediction Problems". In: *Transactions of the ASME, Series D, Journal of Basic Engineering*. D 82, pp. 35–45 (cited in pages 66, 67).

— (1963). "New Methods in Wiener Filtering Theory". In: *Proceedings of the 1st Symposium on Engineering Applications of Random Function Theory and Probability*. New York: John Wiley & Sons, Inc. (cited in page 66).

Kalman, R. E. and R. Bucy (1961). "New Results in Linear Filtering and Prediction Theory". In: *Transactions of the ASME, Series D, Journal of Basic Engineering*. D 83, pp. 95–107 (cited in page 66).

Kamal, M., J. Imura, A. Ohata, T. Hayakawa, and K. Aihara (2012). "Control of Traffic Signals in a Model Predictive Control Framework". In: *Proceedings of the 13th IFAC Symposium on Control in Transportation Systems*. Sofia, pp. 221–226 (cited in page 38).

Kimber, R. M. and E. M. Hollis (1979). “Traffic Queues and Delays at Road Junctions”. In: *TRRL laboratory report, Transport and Road Research Laboratory* 909.1, pp. 1–43. ISSN: 0305-1293 (cited in pages 11, 58).

Kirk, D. E. (2004). *Optimal Control Theory: An Introduction*. Dover Publications. ISBN: 0-486-43484-2 (cited in page 30).

Kraus, W., F. de Souza, R. Carlson, M. Papageorgiou, L. D. Dantas, E. Camponogara, E. Kosmatopoulos, and K. Aboudolas (2010). “Cost Effective Real-Time Traffic Signal Control Using the TUC Strategy”. In: *Intelligent Transportation Systems Magazine, IEEE* 2.4, pp. 6–17. ISSN: 1939-1390 (cited in pages 3, 20).

Kwakernaak, H. (1972). *Linear Optimal Control Systems*. New York, NY, USA: John Wiley & Sons, Inc. ISBN: 0471511102 (cited in page 97).

Le, T., H. L. Vu, Y. Nazarathy, Q. B. Vo, and S. Hoogendoorn (in press 2013). “Linear-quadratic Model Predictive Control for Urban Traffic Networks”. In: *Transportation Research Part C: Emerging Technologies*. ISSN: 0968-090X (cited in page 38).

Lee, J. and B. M. Williams (2009). “Development and Testing of a Constrained Optimization Model for Traffic Signal Plan Transition”. In: *TRB 88th Annual Meeting Compendium of Papers*. Washington DC, USA: Transportation Research Board (cited in pages 26, 39).

Lighthill, M. J. and G. B. Whitham (1955). “On Kinematic Waves II — A Theory of Traffic Flow on Long Crowded Roads”. In: *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 229.1178, pp. 317–345. ISSN: 1364-5021, 1471-2946 (cited in page 62).

Lin, S., B. De Schutter, Y. Xi, and H. Hellendoorn (2011). “Fast Model Predictive Control for Urban Road Networks via MILP”. In: *Intelligent Transportation Systems, IEEE Transactions on* PP.99, pp. 1–11. ISSN: 1524-9050 (cited in page 38).

Ming, S. H. (1997). *Uma Breve Descrição do Sistema SCOOT [A Brief Description of the SCOOT System]*. Portuguese. Notas Técnicas NT201. São Paulo, Brazil: Companhia de Engenharia de Tráfego (cited in page 9).

Mirchandani, P. B. and F.-Y. Wang (2005). “RHODES to Intelligent Transportation Systems”. In: *IEEE Intelligent Systems* 20.1, pp. 10–15. ISSN: 1541-1672 (cited in page 13).

Mirchandani, P. B. and K. L. Head (2001). “A Real-Time Traffic Signal Control System: Architecture, Algorithms and Analysis”. In: *Transportation Research Part C: Emerging Technologies* 9.6, pp. 415–432 (cited in page 14).

Mück, J. (2002a). “Schätzverfahren für den Verkehrszustand an Lichtsignalanlagen unter Verwendung haltestellennaher Detektoren [Traffic State Estimation on Traf-

fic Lights through the Use of Detectors Near the Stop Line]". German. In: *HEUREKA '02 — Optimierung in Verkehr und Transport*. 002/72. FGSV — Forschungsgesellschaft für Straßen-und Verkehrswesen, pp. 385–399 (cited in page 74).

— (2002b). "Using Detectors Near the Stop-line to Estimate Traffic Flows". In: *Traffic Engineering & Control* 43.11, pp. 429–434. ISSN: 0041-0683 (cited in pages 58–60).

Ogata, K. (2010). *Modern Control Engineering*. Prentice Hall. ISBN: 9780136156734 (cited in page 20).

Papageorgiou, M. (1995). "An Integrated Control Approach for Traffic Corridors". In: *Transportation Research Part C: Emerging Technologies* 3.1, pp. 19–30 (cited in pages 17, 94).

Papageorgiou, M. and G. Vigos (2008). "Relating Time-occupancy Measurements to Space-occupancy and Link Vehicle-count". In: *Transportation Research Part C: Emerging Technologies* 16.1, pp. 1–17 (cited in pages 60, 61).

Pohlmann, T. (2010). "A New Method for Online Control of Urban Traffic Signal Systems". PhD thesis. Brauschweig: Technische Universität Braunschweig (cited in pages 15, 16, 20, 21, 46, 56, 62, 72, 74, 81, 95, 96, 98, 100, 109).

Pohlmann, T. and B. Friedrich (2010). "Traffic Signal Transition in Coordinated Meshed Networks". In: *TRB 89th Annual Meeting Compendium of Papers*. Washington DC, USA: Transportation Research Board (cited in pages 6, 39).

Pooran, F. J., P. J. Tarnoff, and R. Kalaputapu (1996). "RT-TRACS: Development of the Real-Time Control Logic". In: *Proceedings of the 6th ITS America Annual Meeting — Intelligent Transportation: Realizing the Benefits*. Houston, USA: ITS America, pp. 422–430 (cited in page 12).

Pueboobpaphan, R. and T. Nakatsuji (2006). "Real-time Traffic State Estimation on Urban Road Network: The Application of Unscented Kalman Filter". In: *Applications of Advanced Technology in Transportation — Proceedings of the 9th International Conference on Applications of Advanced Technology in Transportation*, pp. 542–547. ISBN: 0784407991; 978-078440799-8 (cited in page 58).

Richards, P. I. (1956). "Shock Waves on the Highway". In: *Operations Research* 4.1, pp. 42–51. ISSN: 0030-364X, 1526-5463 (cited in page 62).

RiLSA (2010). *Richtlinien für Lichtsignalanlagen [Guidelines for Signal Controlled Intersections]*. German. FGSV — Forschungsgesellschaft für Straßen-und Verkehrswesen. ISBN: 978-3-941790-02-5 (cited in pages 17, 20, 24).

Robertson, D. and R. Bretherton (1991). "Optimizing Networks of Traffic Signals in Real Time — The SCOOT Method". In: *Vehicular Technology, IEEE Transactions on* 40.1, pp. 11–15. ISSN: 0018-9545 (cited in page 8).

Sen, S. and K. L. Head (1997). “Controlled Optimization of Phases at an Intersection”. In: *Transportation Science* 31.1, pp. 5–17. ISSN: 0041-1655, 1526-5447 (cited in page 15).

Shelby, S., D. Bullock, and D. Gettman (2006). “Transition Methods in Traffic Signal Control”. In: *Transportation Research Record* 1978.1, pp. 130–140. ISSN: 0361-1981 (cited in pages 6, 39, 56).

Sims, A. and K. Dobinson (1980). “The Sydney Coordinated Adaptive Traffic (SCAT) System Philosophy and Benefits”. In: *Vehicular Technology, IEEE Transactions on* 29.2, pp. 130–137. ISSN: 0018-9545 (cited in page 10).

Tampère, C. and L. Immers (2007). “An Extended Kalman Filter Application for Traffic State Estimation Using CTM with Implicit Mode Switching and Dynamic Parameters”. In: *Proceedings of the 10th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 209–216. ISBN: 1-4244-1396-%07 (cited in pages 58, 67).

van der Merwe, R. (2004). “Sigma-point Kalman Filters for Probabilistic Inference in Dynamic State-space Models”. PhD thesis. Oregon Health & Science University (cited in page 104).

van Zuylen, H. J. and L. G. Willumsen (1980). “The Most Likely Trip Matrix Estimated from Traffic Counts”. In: *Transportation Research Part B: Methodological* 14.3, pp. 281–293. ISSN: 0191-2615 (cited in pages 11, 16).

Vigos, G. and M. Papageorgiou (2010). “A Simplified Estimation Scheme for the Number of Vehicles in Signalized Links”. In: *IEEE Transactions on Intelligent Transportation Systems* 11.2, pp. 312–321. ISSN: 1524-9050 (cited in pages 58–60, 62).

Vigos, G., M. Papageorgiou, and Y. Wang (2008). “Real-time Estimation of Vehicle-count within Signalized Links”. In: *Transportation Research Part C: Emerging Technologies* 16.1, pp. 18–35. ISSN: 0968-090X (cited in page 60).

Viti, F. and H. J. van Zuylen (2004). “Modeling Queues at Signalized Intersections”. In: *Transportation Research Record* 1883.8, pp. 68–77 (cited in page 57).

Wan, E. A. and R. van der Merwe (2001). “The Unscented Kalman Filter”. In: *Kalman Filtering and Neural Networks*. Wiley, pp. 221–280 (cited in pages 68, 103).

Wang, Y.-P. (2008). “Optimierung der Quelle-Ziel-Matrixschätzung hinsichtlich Redundanzstörung sich verändernder Verkehrszustände [Optimizing Origin-Destination Matrix Estimation with Respect to Redundant Information and Traffic Conditions]”. German. PhD thesis. Hanover, Germany: Leibniz Universität Hannover (cited in page 16).

Webster, F. V. (1958). *Traffic Signal Settings*. Road Research Technical Paper 39. London, UK: H.M. Stationery Office (cited in pages 17, 20).

Wiener, N. (1949). *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. Cambridge, Massachusetts: The MIT Press. ISBN: 0262730057 (cited in page 66).

Yu, G., J. Hu, C. Zhang, L. Zhuang, and J. Song (2003). "Short-term Traffic Flow Forecasting Based on Markov Chain Model". In: *Proceedings of the Intelligent Vehicles Symposium*. IEEE, pp. 208–212 (cited in page 57).

Zheng, W., D. Lee, and Q. Shi (2006). "Short-term Freeway Traffic Flow Prediction: Bayesian Combined Neural Network Approach". In: *Journal of Transportation Engineering* 132.2, pp. 114–121 (cited in page 57).

List of Symbols

		Page(s)
c_j	Degree of saturation associated to the critical traffic demand being served during stage j, [-]	23
$C^{\max}$	Maximum cycle, [s]	26
$C^{\min}$	Minimum cycle, [s]	26
C^{nom}	Nominal cycle, [s]	20
C^{ref}	Reference cycle, [s]	20
$C^{\text{SAT}}_{\mathcal{J}_n}$	Saturation based cycle for junction n, [s]	24
$C^{\text{trans}}_{\mathcal{J}_n}$	Transition cycle for junction n, [s]	27, 37
$C^{\text{WEB}}_{\mathcal{J}_n}$	Webster's cycle for junction n, [s]	21
$\mathcal{I}_{p_z}$	Set containing the interstages encompassed by phase z	37
$\mathcal{J}$	Set containing all junctions	21
$\mathcal{J}_n$	Junction n	21
K^{cont}	Controller gain, [s]	20
l^{cell}_i	Length of the i^{th} cell, [m]	62
L_i	Interstage of stage i, [s]	21
l^{link}_z	Length of link z, [m]	25, 61
$\mathcal{L}_z$	Set containing all the links that converge to link z	29

		Page(s)
$\bar{\lambda}$	Average network load, [-]	20
λ^{nom}	Nominal load, [-]	20
λ_z	Load on link z, [-]	20
n^{lanes}	Number of lanes	61
$\phi^{\text{ref}}_{i,j}$	Offset reference between i and j, [s]	25
$\mathcal{O}$	Set of available offsets	27
$\mathcal{O}_{i,j}$	Set of available offsets between intersections i and j	48
o^{spc}	Space occupancy, [m]	61
o^{time}	Time occupancy, [s]	61
$\mathcal{P}$	Set containing all phases of the network	37
p_i	Phase i or green time for the phase that gives right-of-way to link i, [s]	21, 29
q^{crit}_j	Traffic demand on the critical phase of stage j, [veh/s]	21
q_{p_i}	Traffic demand of phase i, [veh/s]	22
q^{sat}_j	Saturation flow of the link being served by the phase related to q^{crit}_j, or saturation flow of link j, [veh/s]	21, 29
$q^{\text{tot}}_{\mathcal{O}_{i,j}}$	Total traffic demand related to the offsets in $\mathcal{O}_{A,B}$, [veh/s]	48
s^{fix}_z	Fixed length of stage z, [s]	38
s_j	Stage j or green time of the stage j, [s]	21
s^{max}_z	Maximum available green times for stage z, [s]	37
s^{min}_z	Minimum available green times for stage z, [s]	37
$\mathcal{S}$	Set containing all stages of the network	37
$\mathcal{S}^{\text{fix}}$	Set containing all fixed length stages	22, 38
$\mathcal{S}_{\mathcal{J}_n}$	Set containing all stages of junction n	21
$\mathcal{S}'_{\mathcal{J}_n}$	Set containing all stages of junction n that have no fixed length	22

		Page(s)
$\mathcal{S}_{p_z}$	Set containing the stages encompassed by phase z	37
t^{fill}	Fill-up time of a detector (Mück's queue estimator), [s]	59
t^{ref}	Reference fill-up time (Mück's queue estimator), [s]	59
$t^{\uparrow}_{w,z}$	Turning rate from w to z	29, 48
τ	Jam parameter (Mück's queue estimator)	59
v^{bw}	Back-wave speed, [m/s]	25, 75
v_z^{free}	Free flow speed of link z, [m/s]	25, 75
$\hat{x}$	Estimated vehicle queue length, [veh]	60
$x_z^{\max}$	Maximum number of vehicles that fit in link z, [veh]	20
x^{real}	Real queue length, [veh]	59
x_z	Number of vehicles on link z, [veh]	20, 61

List of Figures

1.1 Cycle Progress 5

2.1 Principles of the SCOOT traffic model 8
2.2 Hypothetical queue profile in OPAC 12
2.3 RHODES architecture 14
2.4 Update process 16

3.1 Multiple-stage phase 22
3.2 Transition cycle 27
3.3 Store and Forward Model 28
3.4 Rolling horizon 31
3.5 Simple network example 31

4.1 Relative Offset 41
4.2 Offset example 47
4.3 Mesh network 49
4.4 Secondary offset compression effect 53
4.5 Fixed time offset 54

5.1 Linear fit 60
5.2 CTM cells 63
5.3 Fundamental diagram of the CTM 63
5.4 Unscented Transformation 68
5.5 Queue points at time k 69

6.1 List network 71
6.2 Traffic flow 73
6.3 New detector positioning 74
6.4 Webster and SAT reference cycles 82
6.5 Webster vs. SAT delays 82
6.6 Webster and SAT reference cycles – 115% 83
6.7 Webster vs. SAT delays – 115% 83

6.8 Reference following in intersections $\mathcal{J}_F$ and $\mathcal{J}_G$ 86
6.9 Delays for the Legacy variants 87
6.10 Delays for the Proposed variants 88
6.11 Delays comparison 91

A.1 CTM Merges and Diverges 99
A.2 Cell merge 99
A.3 Limited-inflow merge cases: $S_A + S_B > R_C$ 100
A.4 Cell diverge 101
A.5 Hypothetical reference following cases 108
A.6 Webster and SAT reference cycles 109
A.7 Webster vs. SAT delays 110
A.8 Update periods multiple of 7.5 minutes 110
A.9 Unsynchronized update periods 112

List of Tables

2.1 Incremental SCOOT actions 9

6.1 Fixed time estimation-only results (no noise) 78
6.2 Fixed time estimation-only results (with 20% multiplicative noise) 78
6.3 Fixed Time prediction results (with 20% multiplicative noise) 79
6.4 TUC results (with 20% multiplicative noise) 80
6.5 Simulation results (with 20% multiplicative noise) 81
6.6 Simulation results (with 20% multiplicative noise) 87
6.7 Simulation results 115% (with 20% multiplicative noise) 88
6.8 Simulation results 90

A.1 Comparison of the statistical indicators 108

List of Algorithms

3.1 Transition Cycle Filter 27

4.1 Loop breaker 50

5.1 Queue Scan 65
5.2 Queue Scan for the Offset Control Module 65
5.3 Queue Transfer to CTM 66

A.1 2 : 1 Cell Merge 101
A.2 Standard Kalman Filter 103
A.3 Unscented Kalman Filter 107